

Thirteenth International Workshop on

Agent-Mediated

Electronic Commerce

AMEC 2011

Esther David

Department of Computer Science
Ashkelon Academic College, Israel
astrdod@acad.ash-college.ac.il

Valentin Robu

School of Electronics and Computer Science
University of Southampton, UK
vr2@ecs.soton.ac.uk

Onn Shehory

IBM Haifa Research Lab
Haifa University, Israel
onn@il.ibm.com

Sebastian Stein

School of Electronics and Computer Science
University of Southampton, UK
ss2@ecs.soton.ac.uk

Programme Committee

Bo An, University of Massachusetts, Amherst, USA
Maria Chli, Aston University, UK
John Collins, University of Minnesota, USA
Florin Constantin, Georgia Institute of Technology, USA
Yagil Engel, IBM Research, Haifa, Israel
Shaheen Fatima, Loughborough University, UK
Nicola Gatti, Politecnico di Milano, Italy
Enrico Gerding, University of Southampton, UK
Mingyu Guo, University of Liverpool, UK
Noam Hazon, Carnegie Mellon University, USA
Minghua He, Aston University, UK
Sverker Janson, SICS, Sweden
Radu Jurca, Google, Switzerland
Wolfgang Ketter, Erasmus University, Netherlands
Han La Poutré, CWI, Netherlands
Jérôme Lang, Université Paris-Dauphine, France
Kate Larson, University of Waterloo, Canada
Peter McBurney, King's College London, UK
Jörg Müller, Technische Universität Clausthal, Germany
David Pardoe, UT Austin, USA
Simon Parsons, Brooklyn College, USA
Zinovi Rabinovich, Bar-Ilan University, Israel
Juan Antonio Rodriguez Aguilar, IIIA, Spain
Alex Rogers, University of Southampton, UK
Jeffrey Rosenschein, The Hebrew University of Jerusalem, Israel
Alberto Sardinha, Instituto Superior Técnico, Portugal
David Sarne, Bar-Ilan University, Israel
Ben Kwang-Mong Sim, GIST, South Korea
Perukrishnen Vytelingum, University of Southampton, UK
William Walsh, Google, USA
Dongmo Zhang, University of Western Sydney, Australia

Table of Contents

FULL PRESENTATIONS

A Trust and Reputation Model for Decision Making in Supply Chain Management	1
Yasaman Haghpanah and Marie desJardins	
Acceptance Strategies for Maximizing Agent Profits in Online Scheduling	15
Mengxiao Wu, Mathijs De Weerd and Han La Poutré	
Bundling in Expert Mediated Search	29
Meenal Chhabra, Sanmay Das and David Sarne	
Autonomously Revising Knowledge-Based Recommendations through Item and User Information	43
Avi Rosenfeld, Aviad Levy and Asher Yoskovitz	
Modeling and Evaluating Human-Like Behavior in Autonomous Agents playing the Social Ultimatum Game	57
Yu-Han Chang, Tomer Levinboim and Rajiv Maheswaran	
Non-Cooperative Bargaining with Arbitrary One-Sided Uncertainty	69
Sofia Ceppi, Nicola Gatti and Claudio Iuliano	

SHORT PRESENTATION

Analysis of Stable Prices in Non-decreasing Sponsored Search Auction	84
ChenKun Tsung, HannJang Ho and SingLing Lee	

FULL PRESENTATIONS

A Trust and Reputation Model for Decision Making in Supply Chain Management

Yasaman Haghpanah and Marie desJardins

University of Maryland, Baltimore County
{yasamanhj, mariedj}@umbc.ed

Abstract. Trust is a critical factor for a successful cooperative relationship in real-world environments. Many such environments, including Supply Chain Management (SCM), can be modeled using multi-agent systems. One shortcoming of current SCM models is that their trust models are ad hoc and do not have a strong theoretical basis. As a result, they are unable to model subtleties in agent behavior that can be used to build a more accurate trust model. In this work, we first propose a trust-based decision framework for SCM that considers multiple trust factors, reported observations (reputation), and direct observations. Then, we present a probabilistic approach for modeling reported observations, the Cognitive Reputation model (CoRe), which will be incorporated into our SCM trust model. We will use the proposed SCM trust model to simulate and study supply chain market behavior in the future.

1 Introduction

Almost all societies need measures of trust in order for the individuals—agents or humans—within them to establish successful relationships with their partners. In supply chain management, establishing trust improves the chances of a successful supply chain relationship, and increases the overall benefit to the agents. Supply chain networks have often been modeled in the research literature with multi-agent systems in which the agents need to cooperate with one or more partners. This collaboration becomes more effective when agents have the ability to choose their partners based on the trustworthiness of the candidates. One major shortcoming in previous research on trust in SCM is that the trust-based decision making is not grounded in a formal trust model.

In modeling trust, there are many sources of information; two of the most important ones are *direct observations* and *reported observations*. Direct observations are beliefs that have been gained by direct interactions with agents; reported observations are acquired by asking other agents. In general, direct observations are more reliable but can be expensive and time-consuming to obtain, while reported observations are cheaper and more readily available but often less reliable.

Reported observations or *reputation* plays an important role in human societies. Since humans (and agents) have a limited amount of time and energy to experience the world directly, learning about other individuals by asking trustworthy third parties can help them to be more successful in their goals and interactions. When people are asked for their opinions about others (or for others' specific behavior), they reply based on their perceptions of those behaviors. Some people are realistic and honest, truthfully

providing the requested information. Others tend to hide the defects of others because they gain personal or economic rewards or incentives by doing so. Still others may change the results with pessimism: for example, when someone is reporting the behavior of a direct competitor and wishes to discredit them. We can also see the results of this changing information in market applications such as Supply Chain Management (SCM), in which organizations (e.g., producers) give incentives to other organizations (e.g., customers) for recommending them to a third party. In this case, customers may tend to hide some of the real flaws and defects of their collaborators in order to obtain the promised incentive. On the other hand, organizations may underreport the work or service of a competitor. Therefore, agents need to recognize the reporting behavior of the reporting agent, in order to obtain more accurate information about other agents.

There are several factors or criteria at play in decision making in a supply chain. For example, in a simple buyer/seller relationship, product delivery, product quality, and product price can all be important criteria in the decision making of a buyer in trading an item with that seller. Buyers may have different trust levels in each of these factors, e.g., they may trust the product price given by the seller, but may not trust that seller's delivery time for that product. As we can see, trust is subjective, and can be defined not only for one factor but for multiple context-dependent factors. This is true also in the real world and when the supply chain is more sophisticated. A trust model that incorporates multiple trust factors for supply chain management, and is grounded in decision theory and probabilistic modeling, is missing from the literature.

In this paper, we first propose a trust model for SCM, that incorporates trust factors specific to SCM and a reputation mechanism. Using decision and game theory, our model builds cooperative agents for supply chain management applications with uncertainties and dynamics. Our ultimate goal is to have a complete and sound trust model for SCM with a strong theoretical basis by combining direct and reported observations with SCM-related trust factors in order to adapt it to real-world scenarios.

Second, we present the Cognitive Reputation (CoRe) model that will be incorporated into SCM in our future work. In the CoRe framework, agents first gather information through reported observations, then model their trust level in reporters' behaviors by learning an agent's characteristic behavior in reporting observations. Finally, a CoRe agent interprets the given information, using it effectively even if the reporters are not honest and their reports are based on faulty perceptions or on dishonest reporting. The key benefit of CoRe's interpretation is in its ability to use all of the reported information, including biased or unfair reports. In our experimental results, we show that CoRe identifies other agents' behaviors faster and more accurately than other state-of-the-art trust and reputation models, even when reported information is incorrect.

To complement our model and benefit from direct observations as well as reported observations, we augment CoRe with one of the existing trust models from the multi-agent literature. Harsanyi Agents Pursuing Trust in Integrity and Competence (HAPTIC) [12], a trust-based decision framework, is among the few existing models with a strong theoretical basis: HAPTIC is grounded in game theory and probabilistic modeling. It has been proved that HAPTIC agents can learn other agents' behaviors reliably using direct observations. One shortcoming of HAPTIC is that it does not support reported observations, which is one of the main contributions of our work.

2 The HAPTIC Model

The HAPTIC model allows an agent to predict a partner's actions and use these predictions to decide whether or not to trust that partner. The key insight in HAPTIC is that it separately models trust using two components of *competence* and *integrity*. Competence is modeled as the probability that a given agent will be able to execute an action in a particular situation. Integrity is an agent's attitude towards honoring its commitments, and is affected by the perceived probability of future interactions. This distinction is useful when a partner defects, because it permits the other agent to determine whether the defection was due to the incompetence of an honest agent, or was the result of cheating by a competent agent with low integrity. HAPTIC identifies a discrete set of player types and maps each agent's competence and integrity θ to a value from this set, which is denoted by Θ . A HAPTIC agent observes the behavior of agents and estimates their competence and integrity, then uses this learned information for decision making in future interactions with that agent.

HAPTIC has been applied to a modified two-player Iterated Prisoner's Dilemma (IPD), in which the payoff matrix in each round is scaled using a random multiplier. As a result, the payoffs differ from one round to the next. HAPTIC assumes that agents know the current round's multiplier before selecting their actions. With variable payoffs, a failure due to low competence can be distinguished from a failure that results from low integrity. An honest but incompetent agent will defect randomly, irrespective of the payoff. By contrast, a cheating agent will show a pattern in its defections that is correlated with the expected payoffs. A HAPTIC agent computes expected payoffs (as defined in the classic Prisoner's Dilemma payoff matrix) and decides rationally whether to cooperate or defect. The agents' interactions are modeled using the Harsanyi transformation from game theory, which converts a game with incomplete information to a strategic game where players may have different types and are uncertain about their opponent's type.

3 Related Work

There has been a significant amount of work in the agent literature on models of trust and reputation. Most of this work incorporates some aspect of direct observations [7]. Fullam and Barber [4] describe a trust decision strategy, including decision making based on both who to trust and how trustworthy to be in reputation exchange networks. Abdul-Rahman and Halles [1] build a system of trust degrees, weightings, and trust operators to merge multiple recommendations. Their scheme assigns higher values to more trustworthy agents, but is not rooted in probabilistic modeling. Sabater and Sierra [11] propose a multi-faceted reputation model; however, their model, REGRET, is neither utility-based nor probabilistic. Instead, it uses ad hoc thresholds and weights.

Interpretation of information in reputation exchange has also been explored. Various models have been developed to interpret the information in reputation reports, including BRS [15] and TRAVOS [13]. Both approaches construct Bayesian models, using the number of satisfactory and unsatisfactory interactions with the sellers as ratings, and use outlier detection or relevance analysis to filter out unreliable ratings. A drawback of

this approach is that a significant amount of information may be considered unreliable, and therefore discarded or discounted. By contrast, CoRe can use biased reports by modeling the specific biases of individual reporters.

BLADE [10] is also a Bayesian model reputation framework; in contrast to BRS and TRAVOS, it does not discard unreliable ratings. Rather, it uses an approach for interpreting unfair ratings. However, this model relies heavily on reported observations, and does not consider direct observations. Additionally, it is not decision-theoretic, and it does not address how to improve payoffs in its interactions with other agents. Vogiatzis et. al. [14] proposed a trust and reputation model that uses a probabilistic framework and focuses on modeling agents whose behavior is not static with time. Their model does not work well in the presence of biased reporters, whereas CoRe is focusing on interpreting biased reputation reports, and therefore works well with different proportions of dishonest and biased opinion providers in the population.

Supply chain markets have been simulated using multi-agent systems [3]. Several approaches have been proposed for adding trust models specifically into SCM. Centotto proposed a reputation mechanism based on organizational concepts and personal norms, with which agents define their preferences about potential interactions [2]. However, this information is not sufficient for adaptively learning trust models, since agents do not model their confidence in the information they receive from other agents. Lin et al. build a trust model based on experiences with suppliers [8]; trust is measured in terms of product quality, order-cycle time, and price. They generalize these factors to the abstract concepts of ability, integrity, and benevolence. This model does not use probabilistic decision theory. Other SCM trust factors have been studied as well, although many of them are focused on specific SCM industries. For example, Paterson et al. studied twelve trust factors, identifying three factors that are critical to the horticulture supply chain: shared values, point-of-sale information, and honesty [9].

4 The Approach for SCM

Our SCM model [5] consists of several layers in a supply network, where each layer contains a number of agents. The layers can correspond to suppliers, producers, distributors, or retailers. Each agent in each level connects to some of the agents in neighboring levels to obtain or provide services, ultimately forming a team or “supply chain.” In general, upstream agents provide services (or offers) to adjacent downstream agents, and downstream agents ask for services or send requests-for-quotes (RFQs) to the adjacent upstream agents, as shown in Figure 1 (a). In this model, we use variable payoffs for different services in different environments. Agents in this framework use a utility function to estimate the future reward that would result from working with a potential partner. This utility function is calculated based on the amount of benefit minus the cost of the transaction.

We consider personal criteria or preferences in the team formation process of SCM. Each downstream agent has a list of criteria and preferences for the services or goods that it needs. For example, one downstream agent might need a high-quality material from an upstream agent, with three weighted criteria: quality 70%, price 20%, and time 10%. In this case, the most important factor for the agent is quality. Downstream agents

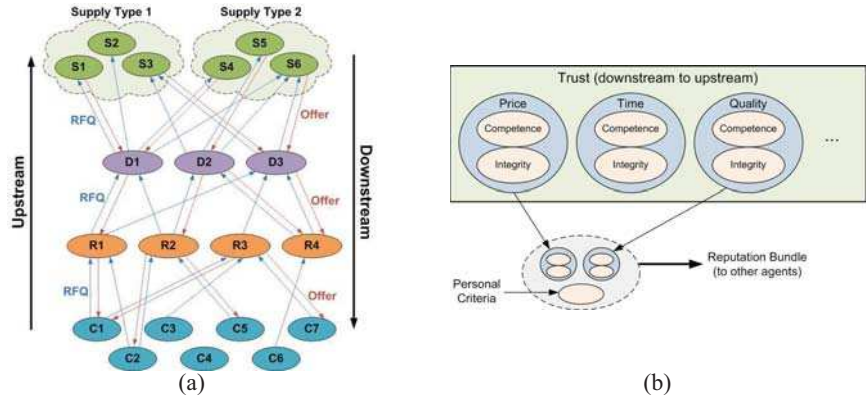


Fig. 1. (a) Example of the SCM model; C, R, D, and S stand for Customer, Retailer, Distributer, and Supplier respectively (b) Trust components and an example of a reputation bundle

send a RFQ to upstream agents. The downstream agents will select the closest match of possible offers based on their criteria and preferences in such a way that the selected offer maximizes the agent’s return utility.

In our model, trust by downstream agents in upstream agents is maximized when the latter agents provide goods and services with low prices and good quality in a timely manner. To model trust in this case, we define the two components of competence and integrity for each factor (e.g., quality, price, and time), as shown in Figure 1 (b). The competence for each of these factors is the probability that the upstream agent is able to fulfill the commitment. Integrity is modeled as the degree to which the agent keeps the same behavior in the long term and in variable-payoff situations. For example, the upstream agent might offer the desired service for two rounds, but after gaining the trust of the downstream agent, the agent might betray in the third round, if they have low integrity for that service. Similarly, the trust of an upstream agent to a downstream agent is affected by the number of times that the downstream agent has accepted the upstream agent’s offer, the payoff level for each interaction, and the frequency of on-time payments. Each of these factors is also modeled using competence and integrity. The combination of these factors will yield an overall trust level of an upstream agent to a downstream agent. An upstream agent can give different offers (on the same trade element) to different downstream agents, since it might have different levels of trust in them based on their competence and integrity. Also, it might accept an RFQ from one downstream agent and not accept the same RFQ from another downstream agent (due to a higher level of trustworthiness in the first agent).

We propose to add another individual-level trust mechanism—namely, reputation exchange—into our model. However, agents might have different opinions and perceptions about the reputation of other agents, and may not report their original observations. To address this complexity, we propose to use CoRe, which is a reputation model that allows for correction of the received reports, as described in the next section.

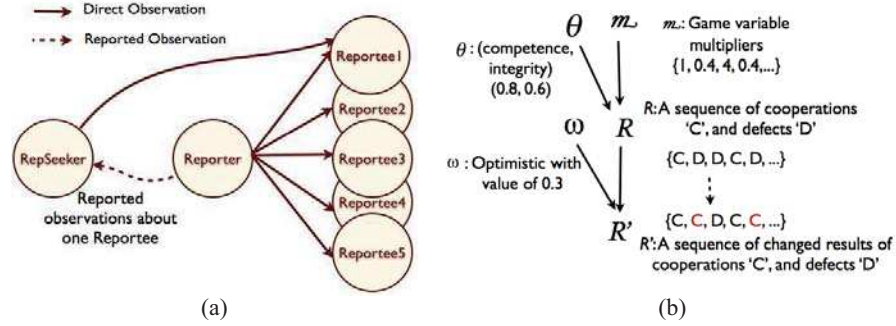


Fig. 2. (a) Basic scenario, (b) Report generation

5 The CoRe Model

In this section, we present a scenario that will be used to define the roles of agents in our model. We then explain how agents learn the reporters' report-generation mechanism, and show how agents use that learned mechanism to correctly interpret later reports. In Figure 2 (a), *RepSeeker* is new to a society of agents. *Reporter* has been in this society for some time and has had direct interactions with several agents (referred to as *Reportee1*, *Reportee2*, *Reportee3*, etc.). *RepSeeker* first starts to interact with *Reportee1* directly, then asks *Reporter* for some information about *Reportee1*. By comparing its own experience to the reported observations of *Reportee1*, *RepSeeker* learns *Reporter*'s reporting behavior (i.e., honest, optimistic or pessimistic). At this point, *RepSeeker* can interpret the acquired reports about other agents (e.g., *Reportee2*) and can use this information to interact more effectively with those agents.

Our trust model incorporates two components: (1) direct observations and (2) reported observations from other agents. As mentioned before, in real-world scenarios, *Reporter* may not always provide correct information about a *Reportee*. Therefore, having a model of *Reporter* and how it generates the reports is needed in order to correctly interpret and use its reports. In order to simulate a more realistic system, we let *Reporter* apply its own perception of the game, change the real report, and then give this new information to *RepSeeker*.

We denote the actual result of the series of games between *Reporter* and *Reportee1* as R (Figure 2 (b)). R is a sequence of Cooperate and Defect actions by *Reportee1* in the series of games played with *Reporter*. CoRe models the interactions and reporting process as follows: *Reportee1* makes its decisions based on its competence and integrity, θ , and the payoff multiplier m of each game, as modeled in HAPTIC [12]. When *Reporter* decides to submit R to *RepSeeker*, it will first change R to R' based on its *Reporter* type, ω , and then deliver R' to *RepSeeker*. For example, if ω is 30% optimistic, then *Reporter* will change the Defects (in R) to Cooperates (in R') with probability 0.3 (Figure 2 (b)).

5.1 Types of Reporters

In this paper, we define three types of Reporters: honest, optimistic, and pessimistic. An honest Reporter always reports truthful information that corresponds directly to

the experience that it has had in the past with other agents. A pessimistic Reporter underestimates other agents’ behavior, and an optimistic Reporter overestimates other agents’ behavior. The level of optimism (or pessimism) is modeled by an ordered pair, $\omega = (\omega_{opt}, \omega_{pess})$, which may be based on the Reporter’s innate characteristic or could depend on Reporter’s incentives for honesty/dishonesty. Specifically, with probability ω_{opt} , Reporter will change some of the Defect actions of the reportee into Cooperates in its reports. Similarly, ω_{pess} defines the probability of changing Cooperate actions into Defects. For optimistic reporters, ω_{opt} represents the degree of optimism (probability of a $D \rightarrow C$ “flip”), and (ω_{pess}) is close to zero. Similarly, for pessimistic reporters, ω_{pess} is the degree of pessimism, and ω_{opt} is close to zero.

A rational agent will be more likely to defect on high-multiplier games and to cooperate on low-multiplier ones. Therefore, an optimistic reporter will preferentially change Defects to Cooperates in rounds associated with high multipliers, conveying a (mistaken) impression that the reportee has been cooperative even in rounds with high multipliers, and a pessimistic agent will be more likely to change Cooperates to Defects in low-multiplier rounds. In our experiments, the RepSeeker first determines the number of “flips” it will make—based on the learned Reporter type—then applies these changes starting with the highest-multiplier rounds (for optimistic) or lowest-multiplier rounds (for pessimistic).

In the real world, a Reporter could have various perceptions of the results of games it plays with different agents, based on its relationship with those players, e.g., as a collaborator or competitor. Here, however, we assume that Reporter has the same perception of different plays, so its reporting behavior will be the same for various agents (i.e., the results will be changed by Reporter in the same pattern for all Reportees). Reporters are also assumed to report the round’s multiplier m honestly, whether they are honest, optimistic, or pessimistic. We intend to relax both of these assumptions in our future work (Section 7).

5.2 Learn Reporter’s Type

RepSeeker can recognize Reporter’s type if it has itself played directly with Reportee1, and has then received a report about Reportee1 from Reporter. Consider our basic scenario, where RepSeeker and Reporter have played separately with Reportee1. Now, the RepSeeker asks Reporter for some information about Reportee1. We denote the actual results of the play between Reporter and Reportee1 by R , and between RepSeeker and Reportee1 by D . Reporter changes the true results, R , based on its type, ω , to R' for reporting to RepSeeker. As in the HAPTIC approach for learning player types, we apply the Harsanyi transformation to learn the Reporter types. We first identify a set of discrete reporter types, Ω .¹ Each type $\omega_i \in \Omega$ is a pair of values $(\omega_{i-opt}, \omega_{i-pess})$. Honest agents are modeled by $\omega_h = (0, 0)$. The probability of a type hypothesis ω_i is denoted by $P(\omega_i)$. RepSeeker has also learned a probability distribution over the possible player types for Reportee1, denoted by θ_j . The probability of each player type is denoted by $P(\theta_j)$. To find the probability of each type of Reporter, given the results R' and D , i.e.,

¹ Using a discrete set of possible agent types is simpler and less computationally expensive than modeling agent types with a continuous variable. We experimented with a continuous version, and the results are very close to what we get with discrete sets.

$P(\omega_i|D', R)$ for each Reporter type, ω_i , we use Bayesian Model Averaging [6] over all possible Reportee types, θ_j :

$$P(\omega_i|R', D) = \sum_{\theta_j \in \Theta} P(\omega_i|R', D, \theta_j) \times P(\theta_j|R', D). \quad (1)$$

The first term, $P(\omega_i|R', D, \theta_j)$, is the probability of a Reporter's type, given Reportee's type θ_j , R' , and D . Since ω_i is conditionally independent of the results of RepSeeker and Reportee's play (D) given θ_j and R' , this term can be simplified to $P(\omega_i|R', \theta_j)$. The second term, $P(\theta_j|R', D)$, is the probability of a Reportee's type, given R' and D . In this case, D , the direct observation, is more reliable than R' , the reported observation. Therefore, CoRe conditions θ_j only on D , and this term is simplified as $P(\theta_j|D)$. We can now rewrite Equation 1 as:

$$P(\omega_i|R', D) = \sum_{\theta_j \in \Theta} P(\omega_i|R', \theta_j) \times P(\theta_j|D). \quad (2)$$

$P(\theta_j|D)$ is RepSeeker's probability distribution of Reportee's type, learned using the HAPTIC model [12]. Using Bayes's rule, we can rewrite the first term of Equation 2 as:

$$P(\omega_i|R', \theta_j) = \frac{P(R', \theta_j|\omega_i) \times P(\omega_i)}{P(R', \theta_j)}. \quad (3)$$

We assume a uniform prior on the Reporter's type, so $P(\omega_i)$ is just the reciprocal of the number of defined types for Reporter ($P(\omega_i) = \frac{1}{|\Omega|}$). Also, $P(R', \theta_j)$ is a normalizing factor, so we only need to compute $P(R', \theta_j|\omega_i)$. Using the definition of conditional probability, this term can be rewritten as:

$$P(R', \theta_j|\omega_i) = P(R'|\theta_j, \omega_i) \times P(\theta_j|\omega_i). \quad (4)$$

We know that θ_j and ω_i are independent, so the second term in Equation 4 is $P(\theta_j)$. Assuming a uniform distribution over the player types, $P(\theta_j) = \frac{1}{|\Theta|}$. The expected value of $P(R'|\theta_j, \omega_i)$ is defined by a weighted sum over all possible values of R :

$$E(P(R'|\theta_j, \omega_i)) = \sum_R P(R'|R, \theta_j, \omega_i) \times P(R|\theta_j, \omega_i). \quad (5)$$

Since computing this full expectation is computationally very expensive, one can instead approximate $P(R'|\theta_j, \omega_i)$ using the maximum likelihood value for R :

$$E(P(R'|\theta_j, \omega_i)) \cong \max_R P(R'|R, \theta_j, \omega_i). \quad (6)$$

Denoting the most likely R as R^* , the term to be maximized on the right-hand side of Equation 6 can be written and expanded as:

$$P(R'|R^*, \theta_j, \omega_i) = P(R'_C, R'_D|R^*, \theta_j, \omega_i), \quad (7)$$

where R'_C are all of the cooperates and R'_D are all of the defects in the report. Since each round played is assumed to be independent of the others, the probabilities of the observed defects and cooperates in the report are independent of each other, yielding:

$$P(R'_C, R'_D|R^*, \theta_j, \omega_i) = P(R'_C|R^*, \theta_j, \omega_i) \times P(R'_D|R^*, \theta_j, \omega_i). \quad (8)$$

Each term in Equation 8 represents a series of i.i.d. (independent and identically distributed) observations from a Bernoulli distribution, so a binomial distribution can be

used to compute the overall probability of each reporter type, based on the number of successes and failures in R'_C or R'_D . A success in this context is a “flip”: that is, when Reporter changes a Cooperate to a Defect, or vice versa. A success in R'_C is therefore defined as observing a Cooperate in R' , in a round in which a Defect was expected in R^* . Conversely, a failure in R'_C is a case where there was a Cooperate in R^* . The same explanation will be applied to Success and Failure in R'_D (i.e., $C \rightarrow D$ flips are successes in this context). The expected success rate for R'_C is the number of $D \rightarrow C$ flips that would be expected from a reporter with type ω_i :

$$\begin{aligned}
E(DC) &= \text{expected number of } (R^* = D \ \& \ R' = C); \\
E(CC) &= \text{expected number of } (R^* = C \ \& \ R' = C); \\
P_{\text{success-}R'_C} &= \frac{E(DC)}{E(DC) + E(CC)} = \\
&= \frac{P(R^* = D|\theta_j) \times \omega_{i_opt}}{P(R^* = D|\theta_j) \times \omega_{i_opt} + P(R^* = C|\theta_j) \times (1 - \omega_{i_pess})},
\end{aligned} \tag{9}$$

and the expected success rate for R'_D will be computed similarly. We calculate the probability of the reported result R' , given ω_i , θ_j , and R^* , using two binomial likelihoods. The first is the probability of observing a certain number of optimistic flips (i.e., the case where the intention R^* of Reportee1 is Defect and the report R' is Cooperate):

$$P[R' = C|R^*, \theta_j, \omega_i] = \text{Binomial}(DC, Total_{R'_C}, P_{\text{success-}R'_C}); \tag{10}$$

The second binomial likelihood is the probability of seeing the observed number of pessimistic flips in the report (when the intention R^* is Cooperate, but is reported as a Defect in R'). This probability is calculated analogously to Equation 10. We multiply these two binomial likelihoods to compute $P(R'|R^*, \theta_j, \omega_i)$ in Equation 8. By averaging over all possible Reportee1 types, RepSeeker can calculate the probability of each type of Reporter (Equation 1). Note that as the number of rounds increases, the statistics become more accurate, leading to better results, as will be shown in Section 6.

5.3 Report Interpretation

In the previous subsection, RepSeeker learned Reporter’s type. In this section, the maximum likelihood of the possible Reporter types will be used to interpret the reported results for new Reportees, since Bayesian model averaging is computationally expensive. We illustrate how agents use this interpretation to learn the player types (competence and integrity) of other agents with whom they have not previously interacted.

After learning Reporter’s type, RepSeeker asks Reporter for information about Reportee2, and uses its learned knowledge of Reporter’s type to interpret the reported results (which are denoted by R'_2). RepSeeker replays the interpreted report using HAPTIC, providing an initial model of Reportee2’s player type before beginning direct interactions with Reportee2. As a result, RepSeeker will have more information about Reportee2 when direct interaction begins than in the first scenario, and this knowledge will increase its overall payoffs in the game.

Without loss of generality, we explain how interpretation works when Reporter’s type is optimistic. Recall that ω_{opt} represents the probability of optimistic flips in the

report and ω_{pess} represents the probability of pessimistic flips in the report. An “Interpret” function estimates the total number of Cooperates, $count_{R_{2C}}$ in the actual results R_2 , using $count_{R'_{2C}}$, as the total number of reported Cooperates in the sequence R'_2 , $rounds$ as the number of rounds in reported play, and ω_{opt} , as shown in Equation 11. The difference between $count_{R_{2C}}$ and $count_{R'_{2C}}$ is the number of Cooperates that should be changed back to Defects to produce more accurate results. The reports is then re-interpreted by changing $(count_{R'_{2C}} - count_{R_{2C}})$ Cooperates in R'_2 , and saving the result as R_2^* . Again, RepSeeker will assume that optimistic reporters have preferentially changed high-multiplier rounds.

Now we play back the new results R_2^* . RepSeeker generates an action as it would do if it were actually playing with Reportee2. Reportee2’s action is denoted by R_2^* . Based on this “most likely” action, HAPTIC can be used to update $P[\theta_j]$ for each possible player type. This distribution will continue to be updated in the online learning process between RepSeeker and Reportee2, when they start their direct interactions.

$$\begin{aligned}
 &count\ cooperations\ in\ R'_2 = count\ coops\ in\ R_2 + \omega_{opt} \times count\ coops\ in\ R_2 \\
 &count_{R'_{2C}} = count_{R_{2C}} + \omega_{opt} \times (rounds - count_{R_{2C}}) \\
 &count_{R_{2C}} = \frac{count_{R'_{2C}} - (\omega_{opt} \times rounds)}{1 - \omega_{opt}}
 \end{aligned} \tag{11}$$

6 Experiments

In this section, we present two sets of experimental results for CoRe framework. In the first set, we use HAPTIC as a baseline, since HAPTIC has been shown to outperform many common strategies in the IPD literature. We also show how uninterpreted reported observations perform, and use it as another baseline (CoRe-NoInterp) in order to show the importance of interpretation of information. A third baseline shows the upper limit of the benefits of reported observations when the reporter is honest (CoRe-Honest). The primary performance metric is the payoff as a function of the number of interactions with an opponent. We also measure the accuracy of the learned Reporter and Reportee1 player types, by looking at the probability assigned to the true player types.

In the second set of experiments, a TRAVOS RepSeeker competes with a CoRe RepSeeker in finding Reportee1’s behavior type. We measure their mean error in finding Reportee1’s type and the cumulative game payoffs. In these experiments, CoRe’s interpretation component uses a discrete set of Reporter types $(\omega_{opt}, \omega_{pess})$. Five types have been considered: (0.3, 0) and (0.7, 0) as optimistic types; (0, 0.3) and (0, 0.7) as pessimistic types; and (0, 0) as an honest reporter.

6.1 HAPTIC Vs. CoRe

In these experiments, we test CoRe in two phases (Figure 3). First, P1 (the RepSeeker) plays 50 rounds and P2 (the Reporter) plays 30 rounds with P3 (the Reportee1). Then, P1 asks P2 about P3. P2 converts the actual results R to R' based on its type ω , and passes the report R' to Player1. Finally, P1 learns the P2’s type, ω , given R' and R using equations in Section 5.2, and uses the learned ω to interpret new reports from P2 about another agent (P4). In the second step, P2 plays 100 rounds with P4 (results

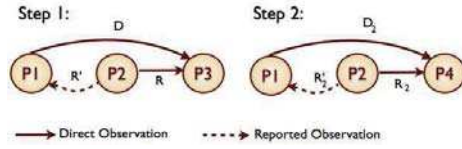


Fig. 3. Step1 and Step2 of basic scenario

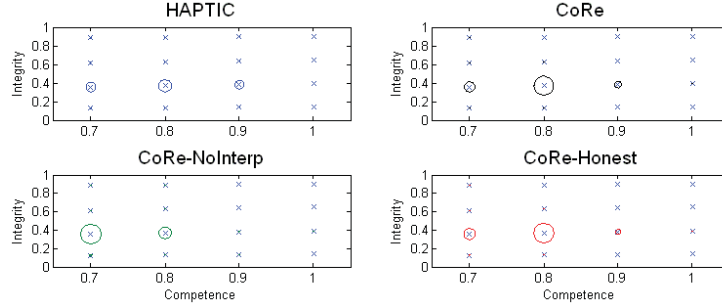


Fig. 4. Learned type (competence and integrity) probabilities in Exp1

= R_2). Then, P1 asks P2 about P4. P2 converts the actual results R_2 to R'_2 based on its type ω , and passes the results to P1. P1 interprets R'_2 based on the learned ω , and generates R_2^* .² P1 plays back R_2^* and learns P4's (C, I) . Now, P1 plays for 20 rounds with P4, starting with its learned values for P4's (C, I) . Using the above scenario, we ran two experiments: Exp1 and Exp2; their settings are listed in Table 1. All results are averaged over 100 runs.

Table 1. Experimental settings for Exp1 and Exp2

	P1	P2	P3	P4	ω_i
Exp1	(0.7,0.9)	(1, 0.8)	(1, 0.3)	(0.8,0.3)	Pess 0.3
Exp2	(0.9,0.9)	random	random	random	random

Figure 4 displays the averaged results of learning Player4's type (competence and integrity) over 100 rounds in Exp1. The possible hypotheses for Player4 are shown by small cross signs; the correct hypothesis is (0.8, 0.35), which is the closest modeled hypothesis to the true value of Player4 (which is (0.8,0.3), as shown in Table 1). The circles' sizes represent the learned probability of each hypothesis for Player4. The top graph from left shows the results for unmodified HAPTIC. In this case, Player1 uses only direct observations. After 20 rounds of play, the hypothesis probabilities are almost equally spread among three values : (0.7, 0.35), (0.8, 0.35), and (0.9, 0.35). Player1 has not yet correctly identified Player4's true type. The CoRe-NoInterp graph shows that using the non-interpreted reports still gives us a moderate probability of finding the correct hypothesis. The results for CoRe are shown in the top graph from right, where it correctly recognizes the true behavior of Player4, and the highest probability is assigned

² Here, R_2^* is P1's estimation of what actually happened between P2 and P4, since this information is not available to P1 as it was in the honest case.

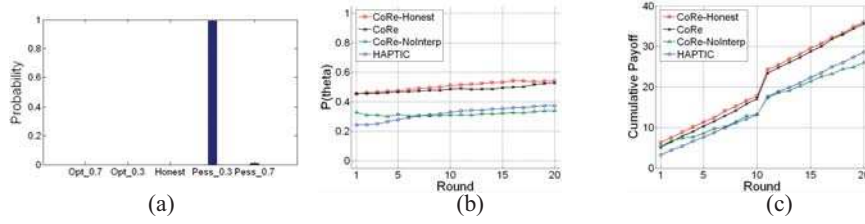


Fig. 5. For Exp1: (a) The probability associated with Player2’s true reported type (Opt 0.7) (b) $P[\theta]$ growth over rounds, and (c) Cumulative payoffs

to (0.8, 0.35). If Player2 were an honest reporter instead of being 30% pessimistic in Exp1, Player1 would have been able to perfectly identify Player4’s actual (C,I) with a high probability, as seen in Figure 4.

Figure 5 (a) shows the results of learning Player2’s ω in Exp1. This graph clearly shows that Player1 was able to identify Player2’s type with a high probability (close to 0.98). This result verifies the correctness of our analysis. Another interesting view of the learning process is how the learned probabilities change over a series of rounds for Player4’s true type. As seen in Figure 5 (b), CoRe starts with a high probability of the true type (near 0.45) from the beginning, while HAPTIC’s probability of the true type remains at a much lower level. The main reason for this behavior is that CoRe has some data from the report it has received from Player2, so it has an initial estimate of Player4’s type. The corresponding payoffs resulting from the four approaches (each for 20 rounds and averaged over 100 runs) are shown in Figure 5 (c). As expected, CoRe with an honest reporter has the highest payoff; CoRe with a 30% pessimistic reporter ranks second, yielding payoffs very close to that of the honest reporter. HAPTIC is in the third place; CoRe without any interpretation is in the fourth place, behaving very similarly to HAPTIC. Since the reporter in this experiment alters Cooperate in the results with only a 30% probability, using reports without interpretations can perform almost as well as HAPTIC (the direct-observation-only approach), but is hindered slightly by its belief in the incorrect reports. Note that the jump in round 11 of the graph is due to the multiplier change of that round to a high-valued one.

Table 2. Payoffs over 100 runs for Exp2

	HAPTIC	No-Interp	CoRe	Honest
Avg-per-run payoff	36.68	41.18	42.67	42.82
Mean-per-round payoff	1.83	2.06	2.13	2.14

To verify the effectiveness of CoRe over different player and reporter types, we performed a second experiment, Exp2, in which we ran a randomized scenario 100 times. In each run, Player1’s type is (0.9, 0.9), and the other players’ and the reporters’ types are selected randomly. The cumulative and mean payoffs for this experiment are reported in Table 2. CoRe achieves 14% improvement over the HAPTIC baseline. A t-test confirms that the mean-per-round of HAPTIC and CoRe are different; with 95% confidence, the difference is between 0.07 and 0.53.

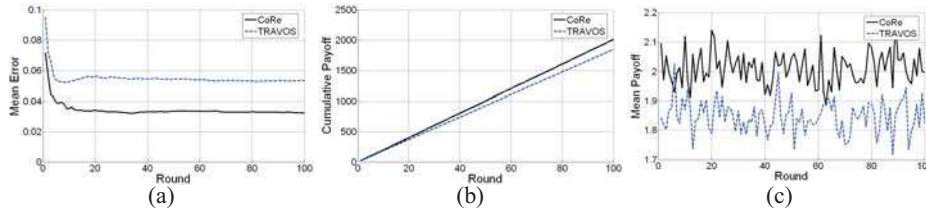


Fig. 6. TRAVOS vs. CoRe: (a) Mean error in identifying correct Reportee’s behavior, (b) Cumulative, and (c) Mean payoffs

6.2 TRAVOS Vs. CoRe

In this subsection, we compare CoRe with TRAVOS [13]. TRAVOS uses probabilistic modeling based on a beta distribution and models both direct and reported observations . It has been shown that TRAVOS outperforms many other trust and reputation models, including probabilistic models like BRS [15]. TRAVOS models the behavior of each agent by a *fulfillment factor*, which is equivalent to “competence” in CoRe. However, TRAVOS does not model the integrity of an agent. To compare TRAVOS to CoRe, therefore, we provide the integrity of an agent as an input to TRAVOS, whereas CoRe is searching in a two-dimensional space for competence and integrity. Note that this gives an advantage to TRAVOS.

We set up another test framework for IPD, Exp3, where a RepSeeker is gaining information from different Reporters to learn the behavior of a Reportee with a randomly selected HAPTIC type. In this experiment, RepSeeker and Reporter’s competences and integrities are fixed at (0.8, 0.9), and the results are averaged over 100 runs. In Exp3, RepSeeker plays with a selected Reportee for 10 rounds. Ten Reporters play for 10 rounds with the same Reportee; the population of these Reporters consists of honest and biased reporters (pessimistic 0.3 and 0.7, and optimistic 0.3 and 0.7). Each Reporter changes the outcome of its play based on its type behavior and then reports the changed results to RepSeeker, which updates its belief about that Reportee.

Despite the fact that we have provided TRAVOS with the correct integrity, as we can see in Figure 6 (a), CoRe outperforms TRAVOS in identifying the Reportee’s type (competence). We show this by the mean error, which is the difference between the identified type and the correct type averaged over 100 runs. This value for TRAVOS has converged to 0.057 and for CoRe to 0.037 (a 35% improvement over TRAVOS). The reason for this discrepancy is that TRAVOS heavily discounts the biased reports, while CoRe interprets and uses that data to learn more about the behavior of the Reportee. As a result of correctly identifying the behavior of the Reporter, the average payoff per round is increased from 1.84 to 2.01 (a 9% improvement), as shown in Figures 6 (b) and 6 (c). The results passed the t-test, which verifies the mean values of HAPTIC and CoRe are different; with 95% confidence, the mean payoff difference is between 0.15 and 0.18.

7 Conclusions and Future Work

In this paper, we presented a proposed trust model to be incorporated into a realistic SCM agent-based model. This proposed work is currently under development. We claim that our model will help to increase (or maximize) the overall profit of the supply chain over time. We also presented a model that can interpret indirect observations provided by reporters in a multi-agent system, based on their learned behavior in previous reputation exchanges. Our experimental results show that a CoRe agent recognizes other agents' behavior more rapidly and accurately than a HAPTIC or TRAVOS agent, and as a result improves its overall payoff. In future, we will investigate how different trust factors affect the system in terms of performance and stability in realistic markets under different conditions. We plan to merge multidimensional trust with CoRe in order to use both direct and reported observations for SCM. We will also improve CoRe by exploring context-dependent reporter types that can cause agents to behave differently in different situations (e.g., when reporting to a competitor versus a collaborator).

References

1. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. In: Hawaii Int. Conference on System Sciences. vol. 33. Maui, Hawaii (Jan 2000)
2. Centeno, R., da Silva, V., Hermoso, R.: A reputation model for organisational supply chain formation. Proc. of the 6th COIN@ AAMAS 9, 33–48 (2009)
3. Collins, J., Ketter, W., Sadeh, N.: Pushing the limits of rational agents: the Trading Agent Competition for Supply Chain Management. *AI Magazine* 31(2), 63 (2010)
4. Fullam, K.K., Barber, K.S.: Learning trust strategies in reputation exchange networks. In: AAMAS-2006. pp. 1241–1248. ACM Press New York, NY, USA (2006)
5. Haghpanah, Y., desJardins, M.: A trust model for supply chain management. In: AAI-10. pp. 1933–1934 (2010)
6. Hoeting, J.A., Madigan, D., Raftery, A.E., Volinsky, C.T.: Bayesian model averaging: A tutorial. *Statistical Science* 14, 382–417 (1999)
7. Huynh, T.D., Jennings, N.R., Shadbolt, N.R.: An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems* 13, 119–154 (2006)
8. Lin, F., Sung, Y., Lo, Y.: Effects of trust mechanisms on supply-chain performance: A multi-agent simulation study. *International Journal of Electronic Commerce* 9(4), 9–112 (2003)
9. Paterson, I., Maguire, H., Al-Hakim, L.: Analysing trust as a means of improving the effectiveness of the virtual supply chain. *International Journal of Networking and Virtual Organisations* 5(3), 325–348 (2008)
10. Regan, K., Poupart, P., Cohen, R.: Bayesian reputation modeling in e-marketplaces sensitive to subjectivity, deception and change. In: AAI-06. vol. 21, p. 1206. AAAI Press (2006)
11. Sabater, J., Sierra, C.: Review on computational trust and reputation models. *Artificial Intelligence Review* 24(1), 33–60 (2005)
12. Smith, M., desJardins, M.: Learning to trust in the competence and commitment of agents. *Autonomous Agents and Multi-Agent Systems* 18(1), 36–82 (February 2009)
13. Teacy, W., Patel, J., Jennings, N., Luck, M., et al.: Coping with inaccurate reputation sources: Experimental analysis of a probabilistic trust model. In: AAMAS-05. pp. 25–29 (2005)
14. Vogiatzis, G., MacGillivray, I., Chli, M.: A probabilistic model for trust and reputation. In: AAMAS-10. pp. 225–232 (2010)
15. Whitby, A., Jøsang, A., Indulska, J.: Filtering out unfair ratings in Bayesian reputation systems. In: Proceedings of the 7th Int. Workshop on Trust in Agent Societies (2004)

Acceptance Strategies for Maximizing Agent Profits in Online Scheduling

Mengxiao Wu¹, Mathijs de Weerd², and Han La Poutré¹

¹ Center for Mathematics and Computer Science (CWI), The Netherlands

² Delft University of Technology, The Netherlands

Abstract. In the market of global logistics, agents need to decide upon whether to accept jobs sequentially offered to them. These jobs, which need to be executed in the near future, have different payments and time constraints. In the offering process we study here, an agent (with limited capacity) needs to make an immediate acceptance decision with little knowledge about future jobs. The goal of the agent is to maximize its profit in such a dynamic environment. We therefore study the online decision problem of acceptance of unit length jobs with time constraints. We consider the problem as a repeated take-it-or-leave-it game which involves online scheduling; we design strategies for when to accept an offered job. Specifically, we present theoretically optimal strategies for a fundamental case, and develop heuristic strategies in combination with an evolutionary algorithm for more general and complex cases. We show experimentally that in the fundamental case the performance of our heuristic solutions is almost the same as that of the theoretical solutions. In various settings, we compare the results achieved by our online solutions to those generated by the optimal offline solutions; the average-case performance ratios are about 1.1. We also analyze the impact of the ratio between the number of slots and the number of jobs on the difficulty of decisions and the performance of our solutions. Although we use a relatively simple scheduling problem to illustrate our approach, we show that it generalizes to online acceptance of jobs in more complex scheduling scenarios as well.

1 Introduction

Consider a market of global logistics in which a large number of jobs are dispatched day and night to many logistics companies. During a period of time, each company gets sequential offers of jobs from the market. Given its limited capacity and time resources, usually, a company can only accept part of the offers. Because of the competition in the market, we suppose the selection decisions are *immediate* and *irrevocable*. The company's target is to maximize its profit through selecting (and executing) jobs. This is an online decision problem, as the company makes the decision on each job offer without prior knowledge of future jobs. To solve it, we make an agent-based model for simulating the decision process of the company (an agent) in the market and design *acceptance strategies* (algorithms) for the agent's optimal decisions.

We first introduce our problem briefly. When job offers arrive one at a time, each job is characterized by a time window for scheduling and a payment. The agent needs to make a take-it-or-leave-it decision immediately. The agent must schedule and execute

every accepted job within its time window so as to get its payment. The utility (profit) that the agent would get is the sum of the payments of all accepted jobs. In our analysis, we assume all jobs have the same processing time, i.e. one time slot, and the agent can execute only one job in each time slot. In this work, we focus on the selection decisions, so we make the scheduling part relatively easy, in which all jobs are assumed to be future activities and no execution happens during the whole offering process.

Our problem may be categorized as a variant of online admission control for interval scheduling [9, 7, 6]. In such problems, the interval between a job's release time and deadline equals the time window in our problem. The authors emphasize the immediate notification of whether to schedule each job at its arrival, which is similar to our selection decision. The decisions in our problem, however, are made at the jobs' offering time, which is not their release time, i.e. the earliest available time for execution. Hence, an accepted job can be rescheduled (within its time window) during the whole offering process. This point distinguishes our problem from almost all online interval scheduling/selection problems in previous work, in which decisions are made at the jobs' release time. Because no job will be executed during the decisions of the jobs that followed, the scheduling part of our problem is more flexible, which increases the complexity of selection. The reason is that with such flexibility in scheduling, the agent has higher expectations of future jobs, but these can also cause him to reject current jobs with good payments that he would otherwise accept.

The agent makes the decision on each job offer in two steps, i) whether this job can be feasibly scheduled together with all previously accepted jobs, and ii) when one or more feasible schedules exist, whether this job is worthy of taking. The focus of this work is on the acceptance strategies rather than the scheduling algorithms. We analyze theoretical solutions in a fundamental case and develop heuristic solutions in general and complex cases. We also present a general idea of using a theoretical analysis of a simple case to determine which are the most important parameters, and then using a machine learning method to find the optimal values of the parameters also in more complex settings. The approach presented in this work can be used to support online decisions in e-commerce applications related to logistics.

Typically, an online solution is evaluated by comparison with an optimal offline solution that knows the entire sequence of jobs in advance. In our experimental analysis, we use an *average-case performance ratio*, which is defined as the ratio between the average result generated by the optimal offline solution and the average result achieved by the online solution on a large number of instances. Our (theoretical and heuristic) solutions generate performance ratios around 1.1 in experiments with various settings. In the fundamental case, the performance of the heuristics is very close to that of the theoretically optimal online solutions. We also analyze the impact of the ratio between the number of slots and the number of jobs. The decision is most difficult when there are two to three times as many jobs as time slots.

The rest of this paper is organized as follows. We first present the problem model in Section 2 and then propose the solutions and acceptance strategies in Section 3. Following the descriptions, in Section 4, the performance of the strategies is evaluated and compared through experiments. Next, we give a brief summary of related work. Finally, conclusion and future work are given.

2 Problem Model

Suppose an agent is offered a finite set N of $n \in \mathbb{N}$ *independent* jobs sequentially. Each job $j \in N$ is characterized by a time window $[x_j, y_j]$ ($x_j, y_j \in \mathbb{N}$) and a payment $z_j \in [0, 1]$, which are independent of each other. Notice that the approach proposed by us works for any given range of payments, but we use the normalized values for ease of presentation. Every job's processing time is one time slot; it must be executed within the given time window. The agent has a set T of $t \in \mathbb{N}$ time slots available for all jobs in N . We let L denote the maximum length of all time windows where $1 \leq L \leq t$, so all jobs' time windows are in T . Given any subset of jobs $A \subseteq N$, we let $\mathcal{S}(A, T) = 1$ denote the existence of one (or more) feasible schedule such that every job $j \in A$ can be uniquely paired with a slot $i \in T$ where $x_j \leq i \leq y_j$. When a new job j is offered, the agent needs to judge whether the set of jobs $A_j \cup \{j\}$ can be feasibly scheduled first, where A_j denotes the set of jobs previously accepted before job j and $\mathcal{S}(A_j, T) = 1$. If $\mathcal{S}(A_j \cup \{j\}, T) = 1$, then the agent needs to make a decision to accept it or not, otherwise the agent can only reject it. Given the set of all accepted jobs $A \subseteq N$ (with $\mathcal{S}(A, T) = 1$), the utility U that the agent would get equals the sum of the payments of all accepted jobs, i.e. $U = \sum_{j \in A} z_j$.

3 Acceptance Strategies

A solution to the problem above is composed of two parts: a scheduling algorithm and an acceptance strategy. For each new job $j \in N$, we consider the scheduling problem of $A_j \cup \{j\}$ as a variant of the Bipartite Matching Problem. All slots T are on one side and all jobs in $A_j \cup \{j\}$ are on the other side; each job only connects to the slots of its time window. A feasible schedule is an one-sided matching in which every job is matched with one slot connected to it. We use the Ford-Fulkerson algorithm [5] to find this kind of matching between jobs in $A_j \cup \{j\}$ and slots in T . If $\mathcal{S}(A_j \cup \{j\}) = 1$, the agent then decides whether to take job j by using acceptance strategies.

We first present two theoretical strategies for a fundamental case in which all jobs have unit time windows; we analyze how to calculate the optimal values of strategy parameters, which maximize the agent's expected utility. Next, we study a general case in which the maximum length of time windows is larger than one: it is very difficult to give analytic solutions for such a setting. Therefore, we develop heuristic strategies for the general case. At last, we give extensions of our strategies for a more complex case in which the precise number of jobs is unknown.

Notice that in the rest of this paper, when we discuss the acceptance decision on a new job j , this is always based on the premise that job j can be feasibly scheduled together with previously accepted jobs in A_j .

3.1 Theoretical Strategies for Unit Time Windows

In this section, we study a fundamental case of the problem, in which every job j 's time window is a single slot denoted by x_j . For theoretical analysis, we assume that the positions of all unit time windows are uniformly distributed on all slots T . We also assume that all jobs' payments are uniformly distributed on the range of $[0, 1]$.

Single Threshold Perhaps the simplest acceptance strategy is setting a single threshold for the payments. If the new job j 's payment is no less than a threshold $\alpha \in [0, 1]$, the agent will accept it. We let $\mathcal{D}_j = 1$ and $\mathcal{D}_j = 0$ denote the agent's acceptance and rejection of job j respectively. The single threshold strategy is given by

$$\mathcal{D}(j) = \begin{cases} 1 & \text{if } z_j \geq \alpha \text{ and } \mathcal{S}(A_j \cup \{j\}) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

We call this the *Theoretical Single Threshold strategy (T1T)*. Next, we present how to determine the theoretically optimal value of α , given the uniform distributions.

We let E^i denote the initially expected utility that the agent would get on each slot $i \in T$; the expected utility on all t slots is $E = t \cdot E^i$. Because t is a constant, the optimal value of α maximizing E^i also maximizes E . As we know, only if at least one job j with $(x_j = i) \wedge (z_j \geq \alpha)$ exists, slot i will finally be occupied by a job; the expected payment of the slot (and the job) is $(1 + \alpha)/2$. The probability of the existence of such job j equals 1 minus the probability that no job has a time window including slot i and a payment of at least α . Reasoning in this way, E^i is given by

$$\begin{aligned} E^i &= P(\exists j, x_j = i \wedge z_j \geq \alpha) \cdot \left(\frac{1 + \alpha}{2}\right) \\ &= \{1 - [P(j \in N, x_j \neq i \vee z_j < \alpha)]^n\} \cdot \left(\frac{1 + \alpha}{2}\right) \\ &= \left[1 - \left(1 - \frac{1 - \alpha}{t}\right)^n\right] \cdot \left(\frac{1 + \alpha}{2}\right) \end{aligned} \quad (2)$$

We can get the optimal value of α by solving formula $\frac{dE^i}{d\alpha} = \frac{1}{2} - \frac{1}{2} \left(1 - \frac{1 - \alpha}{t}\right)^n - \frac{n}{2t} (1 + \alpha) \left(1 - \frac{1 - \alpha}{t}\right)^{n-1} = 0$. In our experiments presented later, we solve it in an approximate way by searching α in $[0, 1]$ with step size of 0.001.

n Thresholds During the whole offering process, the agents may need to make a total of (at most) n decisions: one for each job. In this section, we present a strategy with n thresholds instead of a single threshold for all jobs. If the new job j 's payment is no less than the j^{th} threshold $\alpha_j \in [0, 1]$, the agent will accept it. The strategy is given by

$$\mathcal{D}(j) = \begin{cases} 1 & \text{if } z_j \geq \alpha_j \text{ and } \mathcal{S}(A_j \cup \{j\}) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

We call this the *Theoretical n Thresholds strategy (TnT)*. Notice that the j^{th} threshold α_j is independent of the j^{th} job exactly offered.

We let E_j^i denote the expected utility that the agent would get on an available slot i when the j^{th} job is offered. There are three possibilities. If job j 's slot is slot i and its payment is no less than α_j , which happens with probability $1/t \cdot (1 - \alpha_j)$, the agent will accept it and get an expected payment $(1 + \alpha_j)/2$. Otherwise, if job j 's payment is less than α_j (happening with probability $1/t \cdot \alpha_j$) or its slot is not slot i (happening

Table 1. Simple example

(x_j, z_j)	(2, 0.12)	(1, 0.83)	(3, 0.29)	(3, 0.41)	(2, 0.23)	
<i>T1T</i>	N(< 0.295)	Y(> 0.295)	N(< 0.295)	Y(> 0.295)	N(< 0.295)	$U = 1.24$
<i>TnT</i>	N(< 0.435)	Y(> 0.368)	Y(> 0.282)	N(<i>occupied</i>)	Y(> 0)	$U = 1.35$
Offline	N	Y	N	Y	Y	$U = 1.47$

with probability $(t - 1)/t$, the agent will reject it in the expectation of slot i for the next job $j + 1$. Therefore, the expected utility E_j^i is given by

$$E_j^i = \frac{1}{t} \cdot (1 - \alpha_j) \cdot \frac{1 + \alpha_j}{2} + \frac{1}{t} \cdot \alpha_j \cdot E_{j+1}^i + \frac{t-1}{t} \cdot E_{j+1}^i \quad (4)$$

We calculate the optimal value of α_j by solving formula $\frac{dE_j^i}{d\alpha_j} = \frac{E_{j+1}^i - \alpha_j}{t} = 0$ and get $\alpha_j = E_{j+1}^i$. So if job j 's payment is no less than the agent's expectation of job $j + 1$, given any available slot, the agent will accept it. Otherwise, the agent should leave the slot to job $j + 1$ to get a possibly higher payment. As the expectation of job $n + 1$ is zero, $\alpha_n = 0$. Replacing E_j^i and E_{j+1}^i with α_{j-1} and α_j in Eq. (4) respectively, we get a recursive function $f(j, n, t)$ to calculate threshold α_j where $1 \leq j \leq n$.

$$\alpha_j = f(j, n, t) = \begin{cases} \frac{1}{2t} \cdot (f(j+1, n, t))^2 + \frac{t-1}{t} \cdot f(j+1, n, t) + \frac{1}{2t} & j < n \\ 0 & j \geq n \end{cases} \quad (5)$$

Given a fixed t and a sequence of n , we find that i) for each setting of t and n , threshold α_j is non-linear decreasing; ii) the smaller the n is, the faster the decreasing is; iii) the smaller the n is, the lower the first threshold α_1 is. These match with the intuition that given the same number of slots, the expectations and thus the thresholds decline faster when there are less (future) jobs.

Simple Example Given their definitions, strategy *T1T* is theoretically optimal (in expectation) among single-threshold strategies and strategy *TnT* is theoretically optimal (in expectation) among n -threshold strategies in the fundamental case. We use a simple example with five jobs and three slots to illustrate the differences. In Table 1, the pairs of (x_j, z_j) in the first row represent the jobs in order of arrival (from left to right) where $1 \leq j \leq 5$. In the subsequent rows, the decisions are followed by the thresholds for both of the strategies. Although *TnT* loses some utility on the fourth job by accepting the third one, the advantage of the adaptive thresholds of *TnT* shows in accepting the last job in spite of its relatively low payment.

3.2 Heuristic Strategies

In the fundamental case above, once a job is accepted, its schedule is fixed. However, the length of time windows is generally not unit. The flexibility of (re)scheduling benefits applicability while increasing the difficulty of decisions. In this general case, even if all distributions are still uniform, it is hard to get the optimal values of thresholds in

the above way, given the multiple possibilities of time windows and tremendous possibilities of (re)scheduling. Hence, it is necessary to consider approximate solutions. In this section, we therefore develop heuristic strategies. The basic idea is using multiple parameters to define a decision function; their optimal values are learned by an evolutionary algorithm (EA) [3] through a large number of training sessions.

Single Threshold The first heuristic strategy proposed by us is similar to the theoretical single threshold strategy defined by Eq. (1) except that the optimal value of $\alpha \in [0, 1]$ is determined by the EA. We call this the *Heuristic Single Threshold strategy (H1T)*; its performance is expected to be very close to that of *T1T* in the fundamental case.

n Thresholds Analogously, we also try a heuristic strategy of a different threshold for each job similar to the theoretical one defined by Eq. (3), but let the EA search the optimal combination of the values of those n thresholds $\alpha_j \in [0, 1]$ where $1 \leq j \leq n$. We call this the *Heuristic n Thresholds strategy (HnT)*.

Three Thresholds This strategy divides the whole offering process into three stages by using two parameters $\beta_1, \beta_2 \in [0, 1]$ ($\beta_1 < \beta_2$) and sets a single threshold $\alpha_k \in [0, 1]$ ($1 \leq k \leq 3$) for jobs' payments per stage. The agent will accept job j which is offered in the k^{th} stage only if its payment is no less than α_k . The whole strategy is given by

$$\mathcal{D}(j) = \begin{cases} 1 & \text{if } j \leq \beta_1 \cdot n, z_j \geq \alpha_1, \text{ and } \mathcal{S}(A_j \cup \{j\}) = 1 \\ 1 & \text{if } \beta_1 \cdot n < j \leq \beta_2 \cdot n, z_j \geq \alpha_2, \text{ and } \mathcal{S}(A_j \cup \{j\}) = 1 \\ 1 & \text{if } j > \beta_2 \cdot n, z_j \geq \alpha_3, \text{ and } \mathcal{S}(A_j \cup \{j\}) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

We call this the *Heuristic 3 Thresholds strategy (H3T)*.

Linear Function To be more precise than the strategies with one or three thresholds for payments, we propose heuristic strategies based on *Piecewise Linear Functions* (PLF). As they have fewer parameters to be learned by the EA, it will be easier and faster to find the optimal solutions than the n threshold strategies. The simplest one is a linear function (*PLF1*). We set one parameter α as the slope of the linear function which generates the thresholds for payments, and also set a parameter γ to determine the constant. The agent will accept job j , if its payment is no less than the threshold given by function $p(j)$. The whole strategy is defined by

$$\mathcal{D}(j) = \begin{cases} 1 & \text{if } z_j \geq p(j) \text{ and } \mathcal{S}(A_j \cup \{j\}) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where $p(j) = \alpha \cdot j + \gamma$

To find the global optimum of parameters α and γ for the PLF-based heuristics, we use the EA to learn these within a reasonable range. Any threshold is only reasonable within the range of $[0, 1]$, so $\gamma \in [0, 1]$. Next, given that $j \in \mathbb{N}$ and $z_j \in [0, 1]$, we can derive the range for α as follows.

$$0 \leq \alpha \cdot j + \gamma \leq 1 \text{ and } \gamma \in [0, 1] \implies \alpha \cdot j \in [-1, 1] \implies \alpha \in [-1, 1].$$

Two-piece Piecewise Linear Function The second PLF-based strategy is a two-piece piecewise linear function (*PLF2*). One parameter $\beta \in [0, 1]$ cuts the whole offering process into two stages. The slopes of these two pieces are $\alpha_1, \alpha_2 \in [-1, 1]$ and the constant of the first piece is $\gamma \in [0, 1]$. The agent will accept job j , if its payment is no less than the threshold given by function $p(j)$. The strategy is defined by

$$\mathcal{D}(j) = \begin{cases} 1 & \text{if } z_j \geq p(j) \text{ and } \mathcal{S}(A_j \cup \{j\}) = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{where } p(j) = \begin{cases} \alpha_1 \cdot j + \gamma & \text{if } j \leq \beta \cdot n \\ \alpha_2 \cdot j + (\alpha_1 - \alpha_2) \cdot \beta \cdot n + \gamma & \text{if } j > \beta \cdot n \end{cases} \quad (8)$$

Three-piece Piecewise Linear Function The last one is a three-piece piecewise linear function (*PLF3*). The whole process is divided into three stages by two parameters $\beta_1, \beta_2 \in [0, 1]$ where $\beta_1 < \beta_2$. The slopes of the three pieces are $\alpha_1, \alpha_2, \alpha_3 \in [-1, 1]$. The constant of the first piece is $\gamma \in [0, 1]$. Similarly, the thresholds are still given by function $p(j)$ and the strategy is defined by

$$\mathcal{D}(j) = \begin{cases} 1 & \text{if } z_j \geq p(j) \text{ and } \mathcal{S}(A_j \cup \{j\}) = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{where } p(j) = \begin{cases} \alpha_1 \cdot j + \gamma & \text{if } j \leq \beta_1 \cdot n \\ \alpha_2 \cdot j + (\alpha_1 - \alpha_2) \cdot \beta_1 \cdot n + \gamma & \text{if } \beta_1 \cdot n < j \leq \beta_2 \cdot n \\ \alpha_3 \cdot j + (\alpha_1 - \alpha_2) \cdot \beta_1 \cdot n \\ \quad + (\alpha_2 - \alpha_3) \cdot \beta_2 \cdot n + \gamma & \text{if } j > \beta_2 \cdot n \end{cases} \quad (9)$$

3.3 Dealing with Uncertainty over the Number of Jobs

For the strategies presented above, the number of jobs n is required as an input. We extend the model to a more general case where the total number of jobs is unknown until the whole offering process finishes. Instead of the precise number of jobs n , the agent is only given a range of $[n_{min}, n_{max}]$ and a random distribution. In this work, we assume that n is always uniformly distributed on the range.

For the theoretical strategy *T1T*, it is straightforward to use the expected value of n to calculate the optimal value of the single threshold. This variant of *T1T* is still theoretically optimal. However, we cannot immediately use the expected value of n in the theoretical strategy *TnT*, because the expected value changes after job $j > n_{min}$.

We propose an approximate solution based on *TnT*. We let \bar{n} denote the initially expected value of n , i.e. $\bar{n} = (n_{min} + n_{max})/2$, which is consistent with the offering process until job n_{min} is offered. The agent can calculate threshold α_j by Eq. (5) with input \bar{n} until $j = n_{min}$. After that $j > n_{min}$, the agent's expectation of n is changed by each new offer. We treat the distributions on the range of $[j, n_{max}]$ approximately as uniform distributions. We let \hat{n} denote the average of j and n_{max} , i.e. $\hat{n} = (j + n_{max})/2$. For jobs still coming after job n_{min} , the agent calculates α_j based on \hat{n} instead of \bar{n} . The formal definition is given by

$$\mathcal{D}(j) = \begin{cases} 1 & \text{if } z_j \geq \alpha_j \text{ and } \mathcal{S}(A_j \cup \{j\}) = 1 \\ 0 & \text{otherwise} \end{cases}$$

where

$$\alpha_j = \begin{cases} f(j, \bar{n}, t) & \text{if } j \leq n_{min} \\ f(j, \hat{n}, t) & \text{if } n_{min} < j \leq n_{max} \end{cases}$$

$$\bar{n} = \lfloor \frac{n_{min} + n_{max}}{2} \rfloor, \hat{n} = \lfloor \frac{j + n_{max}}{2} \rfloor \quad (10)$$

where $f(j, n, t)$ is defined in Eq. (5).

To ensure that the heuristics define a threshold for any possible time slot, we replace n by n_{max} in their definitions. By using a representative training set for the EA, the found parameters then incorporate the distribution of n over the range of $[n_{min}, n_{max}]$.

4 Experiments

In the previous sections, we presented two theoretical strategies: $T1T$, TnT and six heuristic strategies: $H1T$, HnT , $H3T$, $PLF1$, $PLF2$ and $PLF3$. In order to evaluate and compare their performance, we set up various experiments. The experimental setting includes the number of jobs n , the number of slots t , the maximum length of time windows L , the random distribution of the starts of time windows x_j , the random distribution of the length of time windows, and the random distribution of payments z_j where $j \in N$. The length of all jobs' time windows is uniformly distributed on the range of $[1, L]$, unless the randomly generated start of the time window plus the maximum length exceeds the slots, i.e. $x_j + L - 1 > t$. In this case, the range is reduced to $[1, t - x_j + 1]$ and the length is uniformly distributed on this new range. The variable settings will be specified when we present the experiments one by one below.

Typically, the performance of online solutions is evaluated by the comparison with the problem's optimal offline solutions. The offline version of our problem is a variant of the Rectangular Assignment Problem, which can be solved by the Hungarian Algorithm [12]. In this work, we use an implementation in MATLAB [4].

Figure 1 illustrates the experimental flow that we follow for each experiment in this work. For instance, given an experimental setting, a theoretical strategy and a heuristic strategy, the experiment will be performed in two stages. First, for the heuristic strategy, use the EA to search the optimal combination of the values of its parameters, given 100 sets of n jobs. Each evaluation of the EA includes 100 simulations based on the 100 instances and the evaluation fitness is defined by the average of the 100 simulation outcomes. As the result, we get an optimal combination of the parameters' values. The heuristic strategy and the optimal values of its parameters form a heuristic solution. Repeat this part 10 times with different sets of 100 instances; 10 heuristic solutions are achieved. Second, cross-evaluate the 10 heuristic solutions by simulations with new 2000 sets of n jobs. The theoretical strategy is also evaluated on the same 2000 instances. To generate benchmarks, we also let the optimal offline solution work on the same 2000 instances in this step.

In this way, for each setting, we get 2000 results of each theoretical strategy and 10×2000 results of every heuristic strategy. We define the performance of a theoretical strategy by the average of the 2000 results. For a heuristic, the average of the 2000

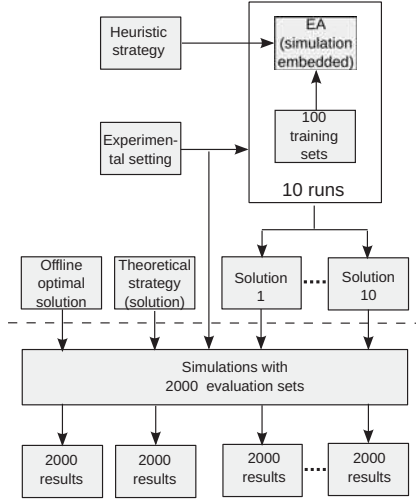


Fig. 1. Experimental flow

results of each solution indicates the solution’s performance. We then define the performance of a heuristic strategy by the average of the 10 averages of different solutions. We also have 2000 results of the optimal offline solution. We define the ratio between the average of the 2000 results achieved by the optimal offline solution and the performance of an online strategy to be the average-case performance ratio, which is no less than 1. The smaller the performance ratio is, the better the online solution performs.

A guideline for the EA’s population size is given as at least $17 + 3 \cdot m^{1.5}$ where m is the number of parameters [3]. The number of parameters of HnT is the same as the number of jobs; the maximum n that we plan to experiment is 110. The numbers of parameters of all the other heuristics are constants: the maximum one is 6. Therefore, we set the population size as 3000 for HnT and set it as 1000 for other heuristics, which are quite sufficient. We also set the EA’s evaluation limit as one million. These settings guarantee that the convergence happens before the evaluation limit is reached, so the (near) optimal results can be found.

4.1 Known Number of Jobs

First, we evaluate all strategies in two cases, unit time windows ($L = 1$) and general time windows ($L \geq 1$), under the environment that the number of jobs n is known. Both cases use the same 7 settings (Table 2); all distributions are uniform distributions.

Unit Time Windows We first compare these strategies in the case of unit time windows, in which strategies $T1T$ and TnT are theoretically optimal. Besides them, the n thresholds strategy is regarded as a more precise one. Therefore, we expect that strategies TnT and HnT will perform best in this case.

Table 2. Experimental settings I

t	n	t/n	L
30	90, 75, 50, 35	0.33, 0.4, 0.6, 0.86	1
15, 50, 70	75	0.2, 0.67, 0.93	1

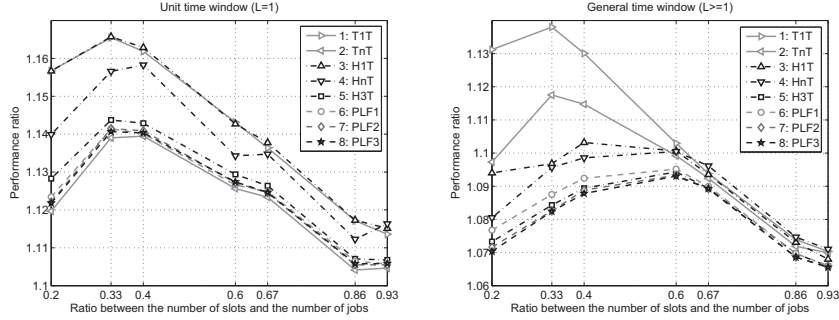
**Fig. 2.** Strategy performance in settings I

Figure 2 (left) illustrates the experimental results. As we expected, the performance of TnT is best in all settings. All PLF-based strategies perform very close to the benchmarks of the online solutions set by TnT ; the three-piece one, i.e. $PLF3$, is the best among them. As the three thresholds of $H3T$ are constants, its performance is slightly worse than that of the PLF-based strategies. Because of the n thresholds, the performance of HnT was expected to be close to that of TnT , but it actually performs worse than the PLF-based strategies and $H3T$ here. One reason is that the size of training sets, i.e. 100 instances per evaluation, is sufficient for other heuristics but is not big enough to prevent over-fitting of the n parameters of HnT . Hence, the results are not optimal in general. By increasing the size of training sets for HnT , the problem can be resolved but the searching time will be significantly extended. The two single threshold strategies perform worst but the largest performance ratio is still small. When the single parameter of heuristic $H1T$ is learned by the EA sufficiently, its performance is almost the same as that of the theoretical strategy $T1T$.

In Figure 2 (left), we notice that the worst performance of all strategies is generated at the point of $t/n = 0.33$; the performance at its right point $t/n = 0.4$ is also low. On one side, when ratio t/n is very close to 1, as the distribution of positions is uniform, each slot is expected to assign one job. The agent's decisions are relatively easy without considering future jobs too much. On the other side, when ratio t/n is very close to 0, each slot is expected to assign many jobs. Because of the uniform distribution of payments, the decisions are also relatively easy: the agent only accepts jobs with very high payments. When the decision problem is easier, the performance of all strategies will be better. The middle area is the most difficult part, in which the agent is indeed in a dilemma between the current job and the expectation/uncertainty of future jobs. Even in this part, however, TnT and $PLF3$ can still generate performance ratios around 1.14.

Table 3. Experimental settings II

t	n	t/\bar{n}	L
30	[70, 110], [70, 80], [35, 65], [30, 40]	0.33, 0.4, 0.6, 0.86	5
15, 50, 60	[60, 90]	0.2, 0.67, 0.8	5

General Time Windows We extend to the case of general time windows. This increases the flexibility of scheduling and also the difficulty of decisions. As our theoretical strategies are derived from the case of unit time windows, their threshold values are no longer optimal in this general case. We still evaluate them here to show the change.

Figure 2 (right) illustrates the experimental results. We notice that when ratio $t/n \geq 0.6$, the performance ratios of all strategies are very close. The reason is, as we mentioned, the decision problem becomes easier in this part as the agent knows that there is a little choice on every slot. On the side of $t/n < 0.6$, the performance of strategies is clearly distinguished. Compared to $T1T$, we find that $H1T$ performs much better, although both of them use a single threshold. This indicates the advantage of the heuristic. By using the EA, the strategy can learn to find the good solutions in various settings. Compared to the results of unit time windows shown in Figure 2 (left), we find that the performance ratios are decreased (so the results are better). We will study the impact of the length of time windows on the performance of the heuristics in our future work. As we expected, the theoretical strategies (derived from the case of unit time windows) perform significantly worse than other heuristics here, because they cannot adapt to the change of the length of time windows.

4.2 Unknown Number of Jobs

The previous experiments evaluated the strategies where the number of jobs n is known. Next, we study the performance of our solutions where n is unknown but uniformly distributed over a given range. Table 3 shows a new set of 7 settings. The expectations of n in all settings correspond to the values of n in Table 2. Although strategy HnT provides good solutions in previous experiments, we omit it in the following experiments with consideration of the cost of experimental time.

Unit Time Windows We compare the strategies under settings with unit time windows and unknown n . As we described, when we use the expectation of n instead of n for $T1T$, the resulting α is still theoretically optimal. Although the approximate variant of TnT is no longer strictly optimal in the theoretical analysis, we think the difference between the approximate solution and the theoretical solution is very small.

Figure 3 (left) illustrates the experimental results. Apparently, TnT still performs best as we expected. The value of the threshold of $T1T$ is still theoretically optimal and the performance of $H1T$ is very close to that of $T1T$. Totally, the performance ratios of all strategies in this case are very similar to those generated in the same settings (except for the issue of n) shown in Figure 2 (left). Considering the increased complexity of the problem, our solutions are robust under dynamic environments.

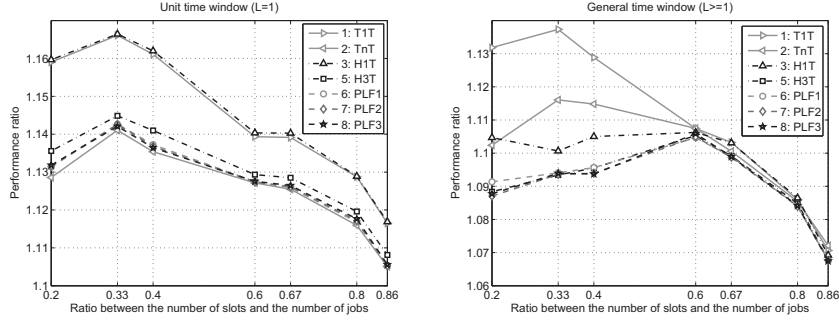


Fig. 3. Strategy performance in settings II

General Time Windows Analogously, we also evaluate the strategies in the case of general time windows and unknown n . Figure 3 (right) illustrates the experimental results, which are quite similar to those shown in Figure 2 (right), except for the results where $t/n = 0.2$. This indicates the robust and adaptive properties of our approach of defining key parameters and using the EA to learn their optimal values.

4.3 Non-uniform Distributions

Further, we evaluate the strategies in more general and complex settings where the random distributions of the starts of time windows and the payments are non-uniform distributions. We experiment various settings, e.g. all payments being exponentially distributed or all starts being normally distributed around a slot close to one end of T ; the resulting average-case performance ratios are between 1.09 to 1.22.

5 Related Work

Our problem relates to the online weighted bipartite matching problem, which is to assign each of sequentially arriving requests to one of the servers given in advance to maximize/minimize the total weight of the matching being produced [10, 11]. Instead of accepting all requests, we focus on selecting a subset of requests to maximize the utility. Thus, our problem is also similar to the multiple secretary problem, which is to select the best m items out of the total $n > m$ items in an online fashion [1]. Instead of the ordinal criterion, Babaioff *et al.* present generalized secretary problems as a framework for online auctions which defines the objective in terms of the numerical values of items [2]. Different from these models, the selection problem studied by us involves a special assignment, i.e. interval scheduling [8]; this combination is also known as an online problem of admission control [9, 6]. Given that jobs arrive online, a scheduler needs to choose whether to schedule each job to maximize the gain. An acceptance notification can either be given when the job really starts or be given once it can be feasibly scheduled. The latter is the same as the requirement of our problem, but our

model permits all accepted jobs to be rescheduled. The scheduling part in our work may be relatively easy, but the online acceptance decision becomes more complex. The reason is that the decision on the current job may influence the decisions about all future jobs in our problem rather than the next few jobs in the problem of interval scheduling.

The problem in [7] is more similar to our work, but the goal is different. They use greedy algorithms, e.g. accepting any job which can be feasibly scheduled (with commitment), and analyze competitive-ratios of these algorithms. We focus on the development of acceptance strategies to maximize the profit rather than the server's utilization and provide exact solutions. Their algorithm called GREEDY can indeed be used for our problem as well and is actually very similar to our single-threshold strategy with a low value. Comparing the resulting average-case performance ratios, on average our other threshold-algorithms perform much better than the GREEDY algorithm.

Summarizing, our model's uniqueness lies in the combination of scheduling and selection, which are influenced by each other during the whole decision process. Our approach also provides a new direction of solving this kind of online decision problem and we evaluate the performance of online solutions by the average-case performance ratio instead of the worst-case competitive ratio.

6 Conclusion and Future Work

In this paper, we have introduced and studied an online decision problem which requires an agent to make acceptance decisions on sequential job offers. The objective is to maximize the utility, the sum of the payments of all accepted jobs. During the whole offering process, the agent's concern is the limited time resources and the expectation of high-payment jobs in the future.

We have presented both theoretical and heuristic solutions. In a fundamental case with unit time windows and uniform distributions, when it is necessary to use the simplest one, our theoretical single threshold strategy $T1T$ can provide the optimal value of the threshold. Our theoretical n threshold strategy TnT can generate the theoretically optimal outcomes in expectation when the number of jobs n is known and still has the best performance amongst all proposed strategies when n is unknown. From fundamental settings to complex settings, compared to the optimal offline solutions, the average-case performance ratios achieved by our online solutions are around 1.1. Overall, the strategy of three-piece piecewise linear function $PLF3$ performs very close to the theoretically optimal online solution in the fundamental case and shows the best performance in all complex settings. As it only has 6 parameters determined by the EA, we say it is a high performance solution which can be specified in a short time. Other heuristics, e.g. $H1T, H3T, PLF, PLF2$, are also very good online solutions requiring even less EA searching time. Even without sufficient training, strategy HnT also generates good results and its performance can be improved if time permits.

Through the experimental analysis, we have pointed out the impact of one key factor, i.e. the ratio between the number of slots and the number of jobs t/n , on the strategy performance. When t/n is at the middle part of $[0, 1]$, the online decision is most difficult. Although the performance of our solutions is a little lower in this part, the performance ratios between 1.09 and 1.22 illustrate the advantage of our solutions for this

dynamic problem. Given various settings, in which it is difficult to find any analytical clue, our solutions show their generality, robustness and adaptivity. Although we make an assumption of unit processing time for all jobs, this work provides an approach that also applies to more generic problems involving both acceptance decisions and complex scheduling. For instance, the heuristic strategies proposed by us could be used in settings with arbitrary length jobs.

Through this work, we have learned that EAs can be used to tune the relevant parameters for settings that are hard to analyze theoretically; this thus gives a general approach, which also works for new settings (although we don't know how good it is in new settings). We answered questions such as i) how to deal with acceptance decisions and scheduling separately, ii) how to find good acceptance strategies, even if it is very hard or impossible to derive an optimal strategy (in expectation) analytically, and iii) which heuristic strategy works best (PLF3), and why (a good balance between accuracy and number of parameters).

In our future work, we would like to derive theoretically optimal solutions for general time windows in addition to our heuristic solutions. Another interesting topic is to extend the problem to a model where the processing time of jobs can vary. We may still use the approach presented in this work but need to add other key factors especially related to scheduling to achieve good results in complex environments. Analysis of competitive-ratios of our algorithms will also be included in our next work.

References

1. M. Ajtai, N. Megiddo, and O. Waarts. Improved algorithms and analysis for secretary problems and generalizations. In *Proc. of the 36th Annual Symposium on Foundations of Computer Science*, pages 473–482. IEEE, 1995.
2. M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg. Online auctions and generalized secretary problems. *ACM SIGecom Exchanges*, 7(2):1–11, 2008.
3. P.A.N. Bosman. On empirical memory design, faster selection of bayesian factorizations and parameter-free gaussian EDAs. In *Proc. of the 11th Annual conference on Genetic and evolutionary computation*, pages 389–396. ACM, 2009.
4. M. Buehren. Algorithms for the Assignment Problem. <http://www.markusbuehren.de>, 2009.
5. L.R. Ford and D.R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8(3):399–404, 1956.
6. S. Fung. Online Preemptive Scheduling with Immediate Decision or Notification and Penalties. *Computing and Combinatorics*, pages 389–398, 2010.
7. J.A. Garay, J.S. Naor, B. Yener, and P. Zhao. On-line admission control and packet scheduling with interleaving. In *Proc. of INFOCOM 2002.*, volume 1, pages 94–103. IEEE, 2002.
8. S. Goldman, J. Parwatikar, and S. Suri. On-line scheduling with hard deadlines. *Algorithms and Data Structures*, pages 258–271, 1997.
9. M.H. Goldwasser and B. Kerbikov. Admission control with immediate notification. *Journal of Scheduling*, 6(3):269–285, 2003.
10. B. Kalyanasundaram and K. Pruhs. On-line weighted matching. In *Proc. of the second annual ACM-SIAM symposium on Discrete algorithms*, page 240. SIAM, 1991.
11. G. Khuller Stephen et al. On-line algorithms for weighted bipartite matching and stable marriages. *Theoretical Computer Science*, 127(2):255–267, 1994.
12. H.W. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Bundling in Expert Mediated Search

Meenal Chhabra, Sanmay Das and David Sarne

Rensselaer Polytechnic Institute,
Troy, NY, USA
Bar-Ilan University
Ramat-Gan, Israel
{chhabm, sanmay}@cs.rpi.edu
sarned@cs.biu.ac.il

Abstract. In search-based markets with noisy signals, like the market for used cars, experts can play an important role. These experts act as information brokers, revealing the true value of a good in exchange for the payment of a fee. Sometimes these experts may choose to sell only bundles of their services (three car inspections for a fixed price, e.g.). We analyze bundling of services in a model of expert-mediated (one-sided) search, and derive optimal strategies for experts and buyers. Our analysis reveals some surprising results. In particular, there are situations where offering only non-unit-size bundles of services can be pareto-improving for the expert and the buyer. Further, in markets with low search costs, the optimal strategy for the expert may well be to offer unlimited services for a single flat fee.

Keywords: One-sided search, Product/service bundling, Agent-mediated search

1 Introduction

Autonomous agents, acting on behalf of their users are commonly required to engage in costly search [5, 8, 13]. The search process is characterized by the need to sequentially evaluate opportunities in order to select one of them. The process of evaluating an opportunity commonly incurs a cost [1], and the searcher’s goal is to maximize the value of the opportunity eventually exploited minus the costs accumulated along the search process. The searcher thus trades off, on each step of its search, the possible marginal gain from revealing the values of further opportunities and the cost associated with doing so [15, 11, 14].

Consider a consumer looking to buy a used car. Typically she will visit potential sellers (or car dealers) in order to estimate the value of the car/s they offer. Visiting sellers and looking at cars incurs a cost (accumulated along the search process), either monetary or in terms of resources that need to be consumed. The search continues until the searcher eventually buys one of the cars seen. While traditional models of sequential search in costly settings assume that the searcher obtains the true value of an opportunity, in many realistic settings she only sees a noisy signal. For example, a non-expert consumer cannot evaluate the mechanical condition of a car. This uncertainty presents an opportunity for

the emergence of knowledge brokers or experts in both traditional and Internet marketplaces. In exchange for the payment of a fee, an expert can disambiguate the noisy signal, providing a better estimate of the true value than the noisy signal available to the non-expert consumer. For example, Carfax.com is a service offering car buyers the history records of any used car, thus enabling a better assessment of its true worth, for a fee. The existence of an expert can theoretically lead to substantially better overall outcomes for buyers [4].

This paper considers what happens when an expert can bundle its services. Bundling of goods and services is a common practice in the real world, ranging from daily necessities at supermarkets to digital information goods and services available online. There is a vast literature on bundling and its advantages and disadvantages from many perspectives [7, 2, 3]. However, research has typically focused on the bundling of complementary and substitute goods. Many studies have looked at profitability and discounts offered in bundling services that use common infrastructure, like phone, cable and internet service, or complementary services like flights, hotels and rental cars.

Bundling has a somewhat different meaning in the context of expert-mediated search. In such markets bundling means that, for a predefined fee, the searcher can use the expert services some fixed number of times. Such schemes are common in physical and virtual markets. For example, Carfax offers a bundle of five reports for \$44.99 (alongside the option to buy a single report for \$34.99). The bundle-based model presented and analyzed in this paper generalizes both the classic sequential search model [11, 10] and its extension to noisy environments. The problem can be formulated as a Stackelberg game, where the expert makes the first move by setting the bundle size and price and the searcher responds. We characterize the searcher’s optimal search strategy, and use that to find the optimal strategy for the expert in terms of bundle size and price. It is not surprising that bundling often leads to an increase in the expected profit of a monopolist expert, but we also find that bundling can be pareto-improving, simultaneously increasing the expected utility of the buyer. While the expert sells bundles at higher prices, the buyer benefits because the extra amount she pays is more than made up for by the higher value of the opportunity eventually picked. We also find that, especially when search costs are low, the profit maximizing strategy for the expert can be to offer unlimited services for a fixed price.

2 The Model

The classic sequential model of one-sided search [15, 11, 14] considers a searcher facing an infinite stream of opportunities from which she needs to choose one. The true values of the different opportunities are initially unknown to the searcher. The searcher is acquainted, however, with the probability density function from which the true values are drawn, denoted $f_v(x)$. The searcher can reveal the true value of an opportunity for a cost c_s . Having no *a priori* information about any specific opportunity, the searcher reviews the opportunities she encounters sequentially. The problem of the searcher is thus to find a strategy S that maps the best value found so far to the decision $\{terminate, resume\}$, given c_s and

$f_v(x)$, in a way that the expected value of opportunity eventually obtained minus the accumulated costs along the search is maximized.

The assumption that the value received is the true value of the opportunity encountered is sometimes relaxed by considering the value obtained to be a noisy signal s , correlated to the true value by a known probability distribution function $f_s(y|v)$ [4, 6, 9]. In these model variants, the searcher usually can obtain, for some additional cost, denoted c_e (e.g., using the services of an *expert* [4], or an interviewing [9] or dating [6] process), the true value (or at least a better estimate) of the opportunity for which signal s is received. In the expert case, which we consider here, we assume the expert can produce its valuation at a cost $d_e \geq 0$, and that the value revealed by the expert is the true value of the opportunity, rather than just a less noisy signal. Here, again, the goal of the searcher is to find a strategy that maps the best signal received so far to the decision $\{\textit{terminate}, \textit{resume}, \textit{query}\}$, where “terminate” means stopping the search, “resume” means evaluating an additional opportunity (without the help of the expert) and “query” means asking the expert to reveal the true value. We consider the case with no recall: if the searcher rejects one opportunity, she cannot go back to it later.¹ Denoting the expected number of times the expert’s services are used by η_q , the expert’s expected profit, π , is given by: $\pi = \eta_q(c_e - d_e)$. The goal of the expert is to find c_e^* that maximizes $\pi(c_e^*)$.

We generalize the model above by introducing the option that experts may offer their services in packages (“bundles”). For a cost c_e^k , the searcher obtains, upon purchasing the bundle, the right to use the expert’s service k times along its search path with no additional cost. The searcher does not necessarily need to make use of all the k queries, and similarly, if she uses them all she can purchase additional bundles as required. The goal of the searcher is to find an optimal strategy as before (mapping from a signal received to $\{\textit{terminate}, \textit{resume}, \textit{query}\}$). When choosing to query the expert the agent needs to pay a cost c_e^k if she has not purchased a bundle yet or if she has already used the expert’s services k times since the last bundle was purchased. When considering the expert’s profit, we need to distinguish between the expected number of bundles purchased by the searcher, denoted η_b , and the expected number of queries used by the searcher, denoted η_q ($\eta_q \leq \eta_b k$). The expert’s expected profit is now given by: $\pi = c_e^k \eta_b - \eta_q d_e$. The goal of the expert is to find the optimal pair (k, c_e^k) that maximizes its overall expected profit π .

3 Analysis and Optimal Policies

We begin by characterizing the optimal strategy of the searcher. Based on the analysis we prove that for a very reasonable assumption on signal structure, namely that “higher signals are good news” as formalized below, the optimal strategy can be represented by a set of reservation values. We then turn to analyzing the expert’s revenue given the bundle characteristics she sets.

¹ This is often the case in real-life settings, e.g., a used car or an apartment seen cannot be guaranteed to remain in the market for long.

3.1 Optimal Search Strategy

Since the searcher cannot recall previous opportunities, her state depends only on the number of remaining pre-paid queries, denoted γ . The system can thus be modeled as a Markov Decision Process with k search states ($\gamma = 0, 1, 2, \dots, k-1$) and one termination state, as illustrated in Figure 1. Upon receiving a signal s when in state $\gamma > 0$, the searcher can either: (a) reject the current opportunity and continue search, starting from the same state γ ; (b) accept the current opportunity and terminate search; (c) query the expert for the true value of the current opportunity and, based on the value received, either accept the current opportunity (terminating the search) or reject the current opportunity and continue search from state $\gamma - 1$. When in state $\gamma = 0$, the searcher has the same options when receiving a signal s , except that querying the expert incurs a cost c_e^k and if the searcher chooses to resume search, based on the value received, she continues from state $\gamma = k - 1$.

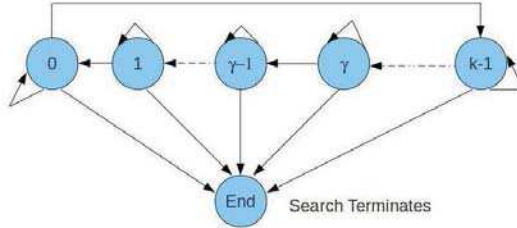


Fig. 1. MDP representation of the searcher's problem (state variable is the number of remaining queries that can be used).

Let V_γ denote the expected value of following the optimal search strategy starting from state γ (the *value-to-go*). For simplicity, we work with the unconditional distribution of signals received, $f_s(x)$, and the distribution of true values conditional on the signals received, $f_v(y|s)$.² First, observe that when the searcher is in state $\gamma > 0$, the only two applicable alternatives are querying the expert and resuming the search without querying the expert.

Proposition 1. *When in state $\gamma > 0$, querying the expert dominates terminating search without querying the expert.*

Proof. For any strategy that terminates search upon obtaining a signal s when in state $\gamma > 0$, consider instead a modification of that strategy which queries the expert, and terminates only if the true value, revealed by the expert, is greater than $V_{\gamma-1}$. This new strategy clearly dominates, since if the true value is less than $V_{\gamma-1}$ the searcher is better off resuming search, and she pays no marginal cost to obtain the expert's service in this instance. \square

For any signal s received in state $\gamma > 0$, the expected benefit if querying the expert, denoted by $M(s, V_{\gamma-1})$, is given by: $M(s, V_{\gamma-1}) = \int_{-\infty}^{\infty} \max(x, V_{\gamma-1}) f_v(x|s) dx$.

² These are interchangeable with the prior distribution of values, $f_v(y)$, and the distribution of signals conditional on values, $f_s(x|v)$, by Bayes' Law.

Therefore, the optimal strategy is to query the expert if $M(s, V_{\gamma-1}) > V_\gamma$. The expected benefit of using the optimal strategy, when starting from state $\gamma > 0$, is thus given by:

$$V_{\gamma>0} = -c_s + V_\gamma \int_{M(s, V_{\gamma-1}) < V_\gamma} f_s(s) ds + \int_{M(s, V_{\gamma-1}) > V_\gamma} f_s(s) M(s, V_{\gamma-1}) ds \quad (1)$$

Similarly, when in state $\gamma = 0$, the expected benefit of obtaining a signal s is: (a) $E[x|s]$ if terminating the search without querying; (b) V_0 if resuming the search without querying the expert; and (c) $M(s, V_{k-1}) - c_e^k$ on querying the expert. For any signal s , the choice which yields the maximum among the three should be made. Let $MAX = \max(E[x|s], V_0, M(s, V_{k-1}) - c_e^k)$, Let ζ_1, ζ_2 and ζ_3 define the sets of signal support such that MAX equals $V_0, E[x|s]$, and $M(s, V_{k-1}) - c_e^k$, respectively. The expected benefit is then:

$$V_0 = -c_s + V_0 \int_{\zeta_1} f_s(s) ds + \int_{\zeta_2} E[x|s] f_s(s) ds + \int_{\zeta_3} f_s(s) (M(s, V_{k-1}) - c_e^k) ds \quad (2)$$

The optimal strategy can be obtained by solving the set of $k - 1$ equation instances of Equation 1 (for $1 \leq \gamma < k$) in addition to Equation 2. An important property of the optimal strategy is given in the following Lemma.

Lemma 1. *When using the optimal strategy, $V_\gamma \leq V_{\gamma+1} \forall 0 \leq \gamma < k - 1$.*

Proof. Suppose $V_\gamma > V_{\gamma+1}$. A searcher starting from state $\gamma + 1$ can follow the optimal strategy as if starting from γ . In this case it will end up with the same value v while the accumulated cost will be at worst equal and possibly less (in the case where search terminates after the last bundle purchase) than the accumulated cost when starting the search from state γ ; therefore the original strategy could not have been optimal. \square

3.2 The HSGN case

While the structure of the optimal strategy tightly depends on $f_s(x|v)$, for some cases, the optimal strategy can have a simple representation in the form of reservation values. For example, suppose the standard assumption that “higher signals are ‘good news’” (HSGN) holds.³ This means that for any $s_1 > s_2$, the conditional distribution of v given s_1 first-order stochastically dominates that of v given s_2 , i.e., $\forall y F_v(y|s_1) < F_v(y|s_2), \forall y$.

Theorem 1. *For $f_v(y|s)$ satisfying the HSGN assumption, for any signal s , the optimal strategy for the searcher in state γ can be described as (see Figure 2):*

(1) a tuple (t_l, t_u, V_{k-1}) , corresponding to state $\gamma = 0$, such that for any signal obtained: (a) the search should resume if $s < t_l$; (b) the opportunity should be accepted if $s > t_u$; and (c) the expert should be queried if $t_l < s < t_u$ and

³ This innocuous assumption loosely means that a higher signal implies a higher probability of a higher true value and is widely used in the literature [12, 16, 4].

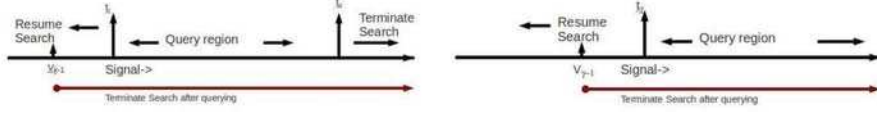


Fig. 2. Characterization of the optimal strategy for noisy search with an expert offering bundles. For state 0 (left figure), the searcher queries the expert if $s \in [t_l, t_u]$ and terminates search if the worth is greater than the value of resuming the search V_{k-1} . The searcher resumes search if $s < t_l$ and terminates without querying the expert if $s > t_u$. However, for any state $\gamma > 0$ (right figure), the searcher rejects the opportunity (terminating search) if $s < t_\gamma$ and otherwise queries the expert before deciding.

the opportunity accepted (and search terminated) if the value obtained from the expert is above the expected value of resuming the search, V_{k-1} , otherwise search should resume; and

(2) a set of $(k-1)$ tuples $(t_\gamma, V_{\gamma-1})$ corresponding to states $\gamma \in 1, 2, \dots, k-1$ such that: (a) the search should resume if $s < t_\gamma$; and (b) the expert should be queried if $s > t_\gamma$ and the opportunity accepted if the value obtained from the expert is above the expected value of resuming the search, $V_{\gamma-1}$, otherwise search should resume.

Proof. (a) The proof augments the one given in [4] regarding the strategy structure in cases where queries are sold only one at a time. We first show that if it is optimal for the searcher to resume search given a signal s , then it must also be optimal for her to do so given any other signal $s' < s$. Then, we show that if it is optimal for the searcher to terminate search given a signal s , then it must also necessarily be optimal for her to do so given any other signal $s'' > s$.

If the optimal strategy given signal s is to resume search then $V_0 > \max(E[x|s], M(s, V_{k-1}) - c_e^k)$. From the HSGN assumption, $E[x|s] > E[x|s']$ (since $s' < s$). Similarly:

$$\begin{aligned}
 M(s, V_{k-1}) &= \int_{x=V_{k-1}}^{\infty} x f_v(x|s) dx + V_{k-1} \int_{x=-\infty}^{V_{k-1}} f_v(x|s) dx \\
 &= V_{k-1} + \int_{x=V_{k-1}}^{\infty} (x - V_{k-1}) f_v(x|s) dx \\
 &> V_{k-1} + \int_{x=V_{k-1}}^{\infty} (x - V_{k-1}) f_v(x|s') dx \\
 &= \int_{x=V_{k-1}}^{\infty} x f_v(x|s') dx + V_{k-1} \int_{x=-\infty}^{V_{k-1}} f_v(x|s') dx \\
 &= \int \max(x, V_{k-1}) f_v(x|s') dx = M(s', V_{k-1})
 \end{aligned}$$

The proof for $s'' > s$ is similar: the expected cost of accepting the current opportunity can be shown to dominate both resuming the search and querying the expert. We omit the details because of space considerations. The optimal strategy can thus be described by the tuple (t_l, t_u, V_{k-1}) as stated in the theorem.

(b) Given Proposition 1, we only need to prove that if, according to the optimal search strategy the searcher should resume her search given a signal s , then she must also do so given any other signal $s' < s$. The proof is similar to (a), ignoring the option to terminate the search without querying the expert, and substituting V_{k-1} with $V_{\gamma-1}$ whenever applicable. \square

Based on Theorem 1, we can construct the appropriate modifications of Equations 1 and 2 for the HSGN case:

$$\begin{aligned}
 V_0 = & -c_s + V_0 F_s(t_l) ds + \int_{s=t_u}^{\infty} f_s(s) E[v|s] ds - c_e^k \int_{s=t_l}^{t_u} f_s(s) ds + \\
 & V_{k-1} \int_{s=t_l}^{t_u} f_s(s) F_v(V_{k-1}|s) ds + \int_{s=t_l}^{t_u} f_s(s) \int_{x=V_{k-1}}^{\infty} x f_v(x|s) dx ds
 \end{aligned} \tag{3}$$

$$\begin{aligned}
 V_{\gamma>0} = & -c_s + V_{\gamma} \int_0^{t_{\gamma}} f_s(s) ds + \int_{t_{\gamma}}^{\infty} f_s(s) \int_{V_{\gamma-1}}^{\infty} x f_v(x|s) dx ds + \\
 & \int_{t_{\gamma}}^{\infty} f_s(s) V_{\gamma-1} F_v(V_{\gamma-1}|s) ds
 \end{aligned} \tag{4}$$

where $F_v(x|s)$ and $F_s(s)$ are the appropriate cumulative distribution functions of $f_v(x|s)$ and $f_s(s)$, respectively.

The values (t_l, t_u) and $t_{\gamma} \forall \gamma > 0$ to be used in the optimal strategy are obtained by deriving Equation 3 with respect to t_l and t_u (separately) and Equation 4 with respect to $t_{\gamma} \forall \gamma > 0$, resulting in (after integration by parts):

$$c_e^k = \int_{y=V_{k-1}}^{\infty} (x - V_{k-1}) f_v(x|t_l) dx \tag{5}$$

$$c_e^k = \int_{-\infty}^{V_{k-1}} (V_{k-1} - x) f_v(x|t_u) dx \tag{6}$$

$$V_{\gamma} = V_{\gamma-1} F_v(\gamma - 1|t_{\gamma}) + \int_{V_{\gamma-1}}^{\infty} x f_v(x|t_{\gamma}) dx \tag{7}$$

Equations 5-7 can intuitively be interpreted as describing the indifference values of the searcher. Equation 5 characterizes the intuition that, at $s = t_l$, the searcher is indifferent between either resuming the search or querying the expert. From equation 6 we see that at $s = t_u$, the searcher is indifferent between either terminating the search or querying the expert. Finally, $s = t_{\gamma}$ in Equation 7 is the signal where the expected gain from rejecting the opportunity with which it is associated is equal to the expected gain received by deciding after querying the expert.

Based on Equations 3-7 we can construct a set of $2k + 1$ equations from which the optimal search strategy can be extracted. These are Equations 3,5-6 for state $\gamma = 0$ and the $2k - 2$ equations resulting from Equations 4 and 7 for $\gamma = 1, 2 \dots k - 1$. We can solve the above system of equations to calculate the equilibrium of the Stackelberg game. The searcher's utility in this case is given by V_0 because the searcher starts from state 0, i.e., with no queries in hand.

3.3 Expert's Perspective

We now turn to formulating the expert's revenue as a function of the bundle characteristics (k, c_e^k) she sets. The expert's revenue is derived in Section 2 by $\pi = c_e^k \eta_b - \eta_q d_e$. Therefore we need to formulate η_b and η_q .

Expected number of bundles purchased The expected number of bundles purchased by the searcher is the expected number of times the searcher queries the expert when in state $\gamma = 0$ (either transitioning to state $\gamma = k - 1$ or terminating the search) (see Figure 1). In order to calculate η_b , we compute the following probabilities:

	Represents	General formulation HSGN formulation
$P_{\gamma \rightarrow \gamma-1}$	Pr (querying and resuming, moving from state $\gamma > 0$ to $\gamma - 1$)	$\int_{M(s, V_{\gamma-1}) > V_\gamma} f_s(s) F_v(V_{\gamma-1} s) ds$ $\Pr(\mathbf{s} > \mathbf{t}_\gamma \text{ and } \mathbf{v} < \mathbf{V}_{\gamma-1})$
$P_{\gamma \rightarrow \gamma}$	Pr (resuming without querying, staying in state $\gamma > 0$)	$\int_{M(s, V_{\gamma-1}) < V_\gamma} f_s(s) ds$ $\Pr(\mathbf{s} < \mathbf{t}_\gamma)$
$P_{\gamma \rightarrow ter}$	Pr (querying and then terminating from state γ)	$\int_{M(s, V_{\gamma-1}) > V_\gamma} f_s(s) (1 - F_v(V_{\gamma-1} s)) ds$ $\Pr(\mathbf{s} \geq \mathbf{t}_\gamma \text{ and } \mathbf{v} \geq \mathbf{V}_{\gamma-1})$
$P_{0 \rightarrow k-1}$	Pr (querying and resuming, moving from state $\gamma = 0$ to $k - 1$)	$\int_{\mathcal{C}_3} f_s(s) F_v(V_{k-1} s) ds$ $\Pr(\mathbf{t}_l \leq \mathbf{s} \leq \mathbf{t}_u \text{ and } \mathbf{v} < \mathbf{V}_{\gamma-1})$
$P_{0 \rightarrow 0}$	Pr (resuming without querying when $\gamma = 0$)	$\int_{\mathcal{C}_1} f_s(s) ds$ $\Pr(\mathbf{s} < \mathbf{t}_l)$
$P_{0 \rightarrow ter}^{-query}$	Pr (terminating without querying when $\gamma = 0$)	$\int_{\mathcal{C}_2} f_s(s) ds$ $\Pr(\mathbf{s} > \mathbf{t}_u)$
$P_{0 \rightarrow ter}^{query}$	Pr (querying and terminating when $\gamma = 0$)	$\int_{\mathcal{C}_3} f_s(s) (1 - F_v(V_{\gamma-1} s)) ds$ $\Pr(\mathbf{t}_l \leq \mathbf{s} \leq \mathbf{t}_u \text{ and } \mathbf{v} \geq \mathbf{V}_{\gamma-1})$

Notice that $P_{\gamma \rightarrow \gamma-1} + P_{\gamma \rightarrow \gamma} + P_{\gamma \rightarrow ter} = 1$, and similarly $P_{0 \rightarrow k-1} + P_{0 \rightarrow 0} + P_{0 \rightarrow ter}^{-query} + P_{0 \rightarrow ter}^{query} = 1$.

Let $P_\gamma(T)$ denote the eventual probability of transitioning, when in state γ , to state $\gamma - 1$ (for $\gamma = 0$ it is the probability of transition to state $k - 1$). Let P_{cycle} be the probability of starting at a given state and getting back to it after going through all other states (excluding search termination state). The values of $P_\gamma(T)$ and P_{cycle} can be calculated as:

$$P_\gamma(T) = \sum_{j=0}^{\infty} (P_{\gamma \rightarrow \gamma})^j P_{\gamma \rightarrow \gamma-1} = \frac{P_{\gamma \rightarrow \gamma-1}}{1 - P_{\gamma \rightarrow \gamma}} \quad ; \quad P_{\text{cycle}} = \prod_{i=0}^{k-1} P_i(T)$$

Now let $P_\gamma(\text{Term})$, denote the probability of terminating the search in current state γ without transitioning to another state and $P_0(\text{Term}|\text{Buy})$ denote the probability of eventually purchasing the bundle and terminating the search in state $\gamma = 0$. These two probabilities are given by:

$$P_\gamma(\text{Term}) = 1 - P_\gamma(T) = \frac{P_{\gamma \rightarrow ter}}{1 - P_{\gamma \rightarrow \gamma}}$$

$$P_0(\text{Term}|\text{Buy}) = \frac{P_{0 \rightarrow \text{ter}}^{\text{query}}}{1 - P_{0 \rightarrow 0}} \quad ; \quad P_0(\text{Term}|\neg \text{Buy}) = \frac{P_{0 \rightarrow \text{ter}}^{-\text{query}}}{1 - P_{0 \rightarrow 0}}$$

Using the above notation, η_b is given by:

$$\begin{aligned} \eta_b &= \sum_{j=1}^{\infty} j \Pr(j \text{ bundles are purchased}) \\ &= \sum_{j=1}^{\infty} j \Pr(j \text{ transitions from state } 0 \text{ to } k-1) \\ &= \sum_{j=1}^{\infty} j P_{\text{cycle}}^{j-1} (P_0(\text{Term}|\text{Buy}) + P_0(T))(1 - P_{\text{cycle}}) \\ &= \sum_{j=1}^{\infty} j \left(\prod_{i=0}^{k-1} P_i(T) \right)^{j-1} (P_0(\text{Term}|\text{Buy}) + P_0(T)) \left(1 - \left(\prod_{i=0}^{k-1} P_i(T) \right) \right) \end{aligned}$$

Expected number of queries used The searcher may terminate the search before exhausting all the remaining queries purchased. This situation is unique to settings where the buyer (searcher in our case) is buying the right to use a service rather than actually receiving the service (or a product) upon purchase. For an expert offering a bundle (k, c_e^k) , the question of how many queries purchased are actually used is important when $d_e > 0$. In order to calculate the expected number of queries used, η_q , we first calculate the probability that exactly m queries are eventually used, denoted by $P_m(Q)$:

$$P_m(Q) = \begin{cases} P_0(\text{Term}|\neg \text{Buy}) & m = 0 \\ P_0(\text{Term}|\text{Buy}) & m = 1 \\ P_0(T) \left(\prod_{j=k-m+2}^{k-1} P_j(T) \right) P_{k-m+1}(\text{Term}) & 1 < m < k \\ P_0(T) \left(\prod_{j=2}^{k-1} P_j(T) \right) P_1(\text{Term}) + P_{\text{cycle}} P_0(\text{Term}|\neg \text{Buy}) & m = k \end{cases}$$

For $m > k$, we represent m as $jk + i$ where $j = \lceil \frac{m}{k} \rceil - 1$ and $i = m - jk$. Here, j is one less than the number of bundles purchased and i represents the number of queries used from the last bundle. This cyclic nature gives us the following recurrence:

$$P_{m=jk+i}(Q) = P_{\text{cycle}} P_{((j-1)k+i)}(Q) = \dots = P_{\text{cycle}}^j P_i(Q) = \left(\prod_{i=0}^{k-1} P_i(T) \right)^j P_i(Q)$$

Therefore the expected number of queries is given by:

$$\begin{aligned} \eta_q &= \mathbb{E}(\text{Number of queries used}) = \sum_{m=0}^{\infty} m P_m(Q) \\ &= \sum_{j=0}^{\infty} \sum_{i=1}^k (jk + i) P_{\text{cycle}}^j P_i(Q) = \sum_{j=0}^{\infty} \sum_{i=1}^k (jk + i) \left(\prod_{l=0}^{k-1} P_l(T) \right)^j P_i(Q) \end{aligned}$$

Once we have calculated η_q and η_b , we can calculate the profit of an expert and find out the optimal strategy for an expert: $\pi = \max_{c_e^k, k} (\eta_b c_e^k - \eta_q d_e)$.

Expected worth of an opportunity received This quantity helps us analyze how much a searcher loses in search cost and query cost. It is also interesting to note how the search cost and query cost affects the expected worth of an opportunity received. Let W be the random variable representing the worth of an opportunity received, then $E_i(W)$ represents the expected worth of the opportunity if the search terminates in state i and $P_i(W)$ represents the probability of terminating at that state.

$$E_i(W) = \begin{cases} \mathbb{E}(v|v > V_{i-1}, s > t_i) & i > 0 \\ \frac{\mathbb{E}(v|v > V_{k-1}, t_l < s < t_u) \Pr(v > V_{k-1} \wedge t_l < s < t_u) + \mathbb{E}(v|s > t_u) \Pr(s > t_u)}{\Pr(v > V_{k-1} \wedge t_l < s < t_u) + \Pr(s > t_u)} & i = 0 \end{cases}$$

$$P_i(W) = \begin{cases} \sum_{j=0}^{j=\infty} P_{\text{cycle}}^j P_0(T) \left(\prod_{h=i+1}^{h=k-1} P_h(T) \right) P_i(\text{Term}) & i > 0 \\ \sum_{j=0}^{j=\infty} P_{\text{cycle}}^j (1 - P_0(T)) & i = 0 \end{cases}$$

$$\mathbb{E}(W) = \mathbb{E}(\text{Expected worth of opportunity}) = \sum_{i=0}^{k-1} E_i(W) P_i(W)$$

Expected number of searches Let η_s be the expected number of searches (opportunities examined by the searcher, $\eta_s \geq \eta_q$). We know the searcher's utility, V_0 , is the value of opportunity received minus the total search and query cost paid in the process ($V_0 = \mathbb{E}(W) - \eta_s c_s - \eta_b c_e^k$). Therefore

$$\eta_s = \frac{\mathbb{E}(W) - \eta_b c_e^k - V_0}{c_s}$$

3.4 Infinite Bundle Size

A specific interesting case to look at is whether the expert would ever want to sell an unlimited supply of services at a fixed price. This is equivalent to an infinite-sized bundle. In this case the model reduces to two states. The searcher starts with state $\gamma = 0$ and continues in this state until she either terminates search or transitions to state $\gamma = \infty$. In the latter case the searcher can keep querying the expert for any reasonable opportunity until she finally finds a sufficiently good opportunity and terminates search (the existence of a search cost ensures that this querying process does not go on forever). Being in state $\gamma = \infty$ is equivalent to being in the world of perfect signals. The optimal reservation value when in state $\gamma = \infty$ can thus be extracted from (e.g., [11]):

$$V_\infty = -c_s + V_\infty \int_0^{V_\infty} f_v(x) dx + \int_{V_\infty}^{\infty} x f_v(x) dx \quad (8)$$

For state $\gamma = 0$, we can use appropriate modifications of Equations 3 and 5-6, replacing V_{k-1} with V_∞ (realizing that the searcher transitions to state $\gamma = \infty$). The optimal strategy can be extracted from solving this set of the four equations.

4 Example

As can be observed from the analysis given in the former section, equilibrium in expert-mediated search with bundling derives from a complex set of dynamics

in the system. The number of parameters affecting the equilibrium is substantial: the distribution of values, the correlation between signals and values, search frictions, the cost of querying the expert, and bundle size, all affect the overall outcome of the process. A static analysis, uncovering phenomenological properties of the model is therefore difficult and restricted. Instead, we turn to a specific example to outline some interesting effects of bundling in this domain.

We illustrate the optimal strategies for the searcher and the expert, assuming some specific distributions of the true values and signals. We consider a case where the signal is an upper bound on the true value. Going back to the used car example, sellers and dealers, offering cars for sale, usually make cosmetic improvements to the cars in question, and proceed to advertise them in the most appealing manner possible, hiding defects using temporary fixes. Specifically, following [4], we assume signals s are uniformly distributed on $[0, 1]$, and the conditional density of true values is linear on $[0, s]$. Then

$$f_s(s) = \begin{cases} 1 & \text{if } 0 < s < 1 \\ 0 & \text{otherwise} \end{cases} \quad f_v(y|s) = \begin{cases} \frac{2y}{s^2} & \text{for } 0 \leq y \leq s \\ 0 & \text{Otherwise} \end{cases}$$

We can use these distributions to solve the systems of equations as described in detail in the previous section. Solving these leads to several interesting insights on the effects of bundling. Surprisingly, we find that forced bundling can be pareto improving. It can lead to improvement not only in the profit of the monopolist expert, which is to be expected, but simultaneously to improvement in the expected utility of the searcher from engaging in the search process. Figure 3 shows the effect and the intuition. The figure depicts the expected utility to the buyer and the expert's expected net profit (alongside the bundle and per-query costs) as a function of the bundle size⁴ for $c_s = 0.01$ and different d_e values. When the marginal cost of producing an expert report is zero (as in the “digital services” case: an extra Carfax report can be produced essentially for free), there is no significant added cost to the expert. The search overall is becoming more efficient, and the expert and the buyer can split the additional utility. On the other hand, for non-zero marginal cost of producing an extra expert report, the benefit to the expert of selling higher bundle sizes rapidly declines after bundle sizes of two and three, because there is a real cost incurred in producing the additional reports that the searcher may demand.

Another interesting observation concerns the correlation between bundling and search cost (the cost of seeing each opportunity initially, c_s) faced by the user. In a world of high search costs, users do not expect to keep searching for more than a few opportunities, so they are unlikely to be willing to purchase a bundle of high size. Interestingly, when the search costs become very low, we find that it can be optimal for the expert to sell an unlimited subscription to her services for a fixed fee. Figure 4 shows an example of this phenomena in the zero marginal cost scenario. A search cost $c_s = 0.001$ yields a setting where it is optimal for the expert to offer infinite-size bundles, but increasing the search cost slightly to 0.003 leads to high (but far from infinite) bundle sizes being

⁴ For each bundle size, the appropriate optimal c_e^k is used.

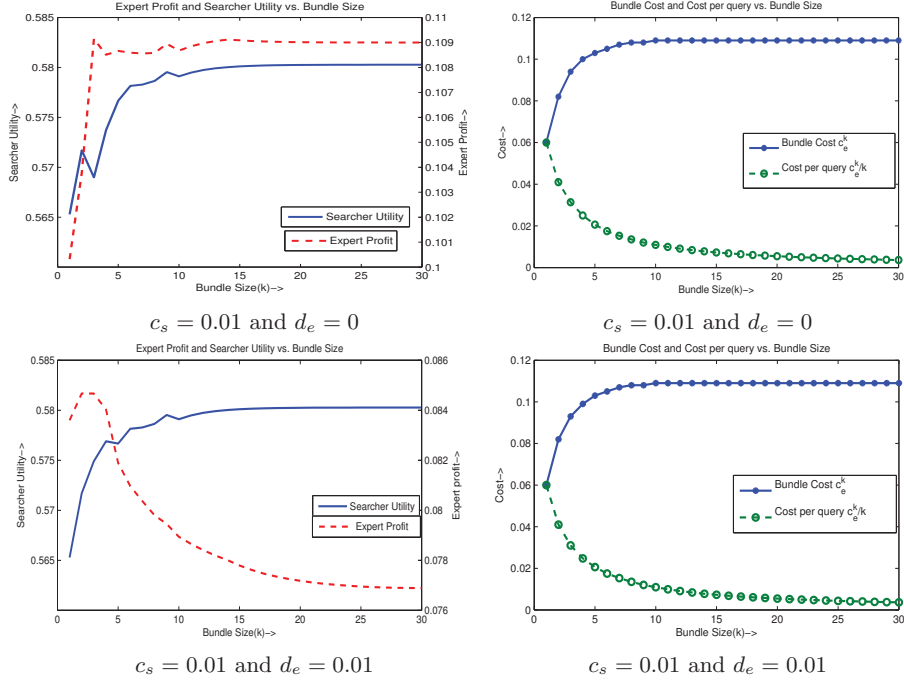


Fig. 3. Effect of bundling on the price charged by the expert and its impact on the searcher’s utility and the expert’s profit. In this case, the searcher’s utility increases as the bundle size increases. We see that although the overall bundle price, c_e^k , increases with the increase in bundle size, the cost per query decreases rapidly.

preferred. Looking back at Figure 3 we observe that when $c_s = 0.01$, the optimal bundle size is actually very small (in fact it is around 2-4 for several examples).

Both these observations, that increasing either the marginal cost of producing expert services and/or the cost incurred by the buyer in searching lead to smaller bundle sizes being preferred, correspond well to the real world. Unlimited subscription models are usually available in online services, where vanishing marginal costs and low search costs dominate (like autocheck.com), whereas traditional marketplaces have smaller bundle sizes.

5 Discussion

Experts can play a significant role in search-based marketplaces where there is a niche for information brokers – this is the premise of mechanics who inspect used cars to make sure they are not lemons and financial experts for due diligence of companies at the service of investors, and more. The ubiquity of electronic markets has changed the nature of many markets not just by lowering the cost of search to the consumer, but also by making it possible for experts to produce additional expert reports and communicate them to the consumer at

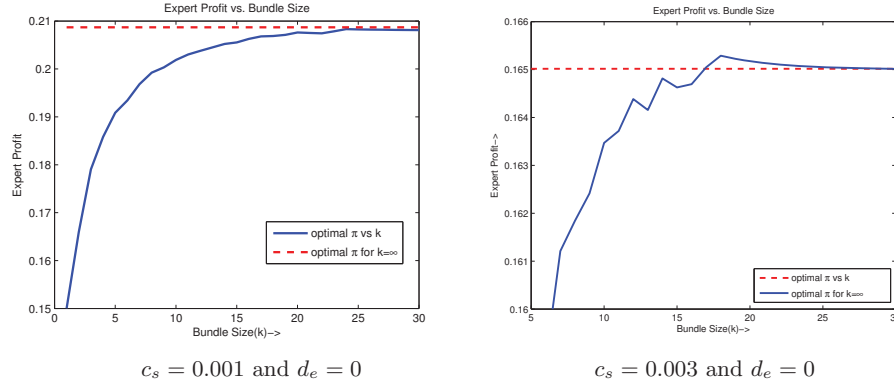


Fig. 4. Expert’s profit for different bundle size for $c_s = 0.001$ with $d_e = 0$. Here, selling infinite size bundle is most profitable i.e., it is optimal for an expert to offer unlimited subscription to her service for a fixed fee.

little to no extra cost. Expert services like Carfax have emerged naturally. An interesting observation is that these agencies have experimented with several different bundling and subscription models. In this paper we provide a model that helps us make sense of these different bundling strategies. While bundling has been studied in similar domains, e.g., Bakos and Brynjolfsson have shown that bundling is particularly useful for digital information goods because of the low marginal cost [2], prior research has not considered the interactions of bundling and search.

Surprisingly, we find that constraining experts to sell only bundles of their services can improve outcomes for both experts (who make higher profits) and searchers (who gain in expected utility of the search process). The intuition is that, by purchasing a bundle, the searcher is less constrained by the marginal cost of expert services and can exploit the search process to find a better opportunity. Another surprising result is that in some circumstances the expert may in fact maximize profit by selling an unlimited subscription to its service, compared with any finite bundle size. These circumstances are characterized by very low search costs and close-to-zero marginal costs of producing an extra unit of the expert’s service, a case which is highly applicable in electronic marketplaces.

Similar to the general bundling literature, we also observe negative correlation between marginal cost and optimal bundle size. In addition, we show that search cost can also affect optimal bundling significantly. Bundling is discouraged if the search cost is high, even if the marginal cost is zero, because buyers will typically not want to sample many opportunities, so the marginal benefit to them of extra expert reports that are essentially “free” is minimized.

There are three major directions for future work. First, our model only allows for bundles of fixed size. Mixed bundling, where the expert sells bundles of different possible sizes, has been shown to perform better in the theory of product bundling, and would be an interesting extension in our domain. Second, our model assumes a monopolistic expert. Bakos and Brynjolfsson have shown

that bundling, even in inferior digital “information” goods, with close to zero marginal cost has the potential to drive away superior quality sellers [3]. However, their analysis does not take search into consideration. Incorporating experts of different quality and cost who compete with each other in our model could reveal new insights. Third, in our relatively simple model, the improvement in utilities is essentially free and social-welfare maximizing, as it is a product of improving the efficiency of search. This is not unrealistic, and such “one-sided” search models have been shown to have great applicability. However, in many real markets, it is also important to consider the utility of the seller, which involves modeling search as a two-sided process. Even in two-sided markets search frictions can play a major role, thus it is important to investigate whether the presence of experts can improve outcomes in such markets.

References

1. Y. Bakos. The emerging role of electronic marketplaces on the internet. *Commun. ACM*, 41(8):35–42, 1998.
2. Y. Bakos and E. Brynjolfsson. Bundling information goods: Pricing, profits, and efficiency. *Management Science*, 45(12):1613–1630, 1999.
3. Y. Bakos and E. Brynjolfsson. Bundling and Competition on the Internet. *Marketing science*, 19(1):63–82, 2000.
4. M. Chhabra, S. Das, and D. Sarne. Expert mediated search. In *Proceedings of the Tenth International Joint Conference on Autonomous Agents and Multi-Agent Systems. To appear*, Taipei, Taiwan, May 2011.
5. S. Choi and J. Liu. Optimal time-constrained trading strategies for autonomous agents. In *Proc. of MAMA ’2000*, 2000.
6. S. Das and E. Kamenica. Two-sided bandits and the dating market. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 947–952, Edinburgh, UK, August 2005.
7. W. Hanson and R. Martin. Optimal bundle pricing. *Management Science*, 36(2):155–174, 1990.
8. J. Kephart and A. Greenwald. Shopbot economics. *JAAMAS*, 5(3):255–287, 2002.
9. R. Lee and M. Schwarz. Interviewing in two-sided matching markets. *NBER Working Paper*, 2009.
10. S. Lippman and J. McCall. The economics of job search: A survey. *Economic Inquiry*, 14:155–189, 1976.
11. J. McMillan and M. Rothschild. Search. In R. Aumann and S. Hart, editors, *Handbook of Game Theory with Economic Applications*, pages 905–927. 1994.
12. P. Milgrom. Good news and bad news: Representation theorems and applications. *The Bell Journal of Economics*, pages 380–391, 1981.
13. S. Parsons and M. Wooldridge. Game theory and decision theory in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 5(3):243–254, 2002.
14. M. Rothschild. Searching for the lowest price when the distribution of prices is unknown. *Journal of Political Economy*, 82(4):689–711, 1974.
15. M. Weitzman. Optimal search for the best alternative. *Econometrica: Journal of the Econometric Society*, 47(3):641–654, 1979.
16. R. Wright. Job search and cyclical unemployment. *The Journal of Political Economy*, 94(1):38–55, 1986.

Autonomously Revising Knowledge-Based Recommendations through Item and User Information

Avi Rosenfeld and Aviad Levy and Asher Yoskovitz

Jerusalem College of Technology, Jerusalem 91160, Israel
rosenfa@jct.ac.il, aviadl@jct.ac.il,
asher.yoskovitz@mysupermarket.com

Abstract. Recommender systems are now an integral part of many e-commerce websites, providing people relevant products they should consider purchasing. To date, many types of recommender systems have been proposed, with major categories belonging to item-based, user-based (collaborative) or knowledge-based algorithms. In this paper, we present a hybrid system that combines a knowledge based (KB) recommendation approach with a learning component that constantly assesses and updates the system's recommendations based on a collaborative and item based components. This combination facilitated creating a commercial system that was originally deployed as a KB system with only limited user data, but grew into a progressively more accurate system by using accumulated user data to augment the KB weights through item based and collaborative elements. This paper details the algorithms used to create the hybrid recommender, and details its initial pilot in recommending alternative products in an online shopping environment.

1 Introduction

Recommender systems have become an integral part of many e-commerce websites, giving consumers suggestions for additional or alternative products to purchase. These systems are part of well known websites such as Amazon.com, Pandora, Yahoo!, and Netflix [2–4, 7]. In fact, Netflix recently offered a Million Dollar Prize [2] for significantly increasing the quality of its recommendations, highlighting the importance of this field to e-commerce websites.

For commercial companies, recommendations are important to both directly and indirectly generate sales. Direct sales can be generated in two ways. First, a person may wish to buy a specific product from a website, but not be able to complete the transaction due to the product no longer being in stock. The recommender system can then provide alternate products, still completing a sale. In a second scenario, even if the product is in stock, the recommendation system may be able to provide additional items that the user may wish to buy, even furthering the revenue from the website. Even if the recommender system does not directly produce sales, they can be critical in providing an improved shopping experience thus attracting more shoppers to the website and indirectly producing more sales. In these types of scenarios, the recommender system can provide additional information about related products or services that might aid the user in better using a product they just purchased. In these types of cases, the recommender

system can provide an after sales support system, ensuring the buyer is satisfied with the purchase.

In this paper, we describe the recommender system we built for the e-commerce website, mysupermarket.co.uk. MySupermarket is a relatively small private e-commerce company that makes its revenues by providing recommendations of grocery products to buy. All revenues are generated as a percentage of the total order places, so it is critical that the shopping experiences be as pleasant as possible, and recommendations be as relevant as possible, to boost sales. One of the key features of MySupermarket is its five ways it helps users save money¹. The first and most important mechanism is a “swap and save” feature where the recommender system provides alternate (swap), yet similar, items to the user that are cheaper (save). This paper focuses on the algorithms involved with the recommender agent in this system.

The novelty of MySupermarket’s swap and save agent lies in its combination of knowledge based, collaborative filtering and item based algorithms. In the next section, we details the background of the recommender algorithms upon which our hybrid system is based, and stress the contribution of this work. In Section 3, we describe MySupermarket’s current recommender agent, which integrates the expert’s knowledge exclusively to produce recommendations. Unique to our system is a learning agent that creates recommendations based on the current expert recommendations, but also autonomously updates the expert’s recommendation with item based and collaborative information. This approach is novel in that it presents the first hybrid of all major types of recommender technologies: knowledge, item based and collaborative. We detail this approach in Section 4. Section 5 concludes and provides directions for how this work can be generally applied to other systems as well.

2 Related Work

To date, two major groups of algorithms have been proposed for use in recommender systems, *collaborative* and *item based* approaches [1, 3, 5, 7, 10]. The term collaborative filtering was coined by the designers of one of the first of these systems, Tapestry [6], to capture that people often obtain information through collaborating with one another to obtain information. Systems based on collaborative approaches (also called user based) have been widely used in many commercial applications [2, 6, 5, 4, 7] and facilitate giving a given user recommendations based on the past behavior of a known group of similar users. A second popular group of recommenders are item based (often called content based) approaches and focus on similarities between items to produce recommendations, typically based on the type of content of the item that is being search for [1, 4, 7, 10]. These approaches assume a generality between all types of users, and focus on shared characteristics between all members of the system. For example, assume preset categories exists for types of genre for books or movies (e.g. comedy, mystery, documentary, and classic). Once we have identified the genre of one item that is being searched for by all users, we can recommend other items of the same type. Theoretically there is no need within this approach to consider a given user’s history once a categorization scheme has been implemented based on the item based approach.

¹ <http://www.mysupermarket.co.uk/Help/FAQ.aspx/>

One major disadvantage in both the collaborative and item based approaches is the time required and / or the needed data required to build these models. This is often referred to as the “cold start” or “ramp up” problems whereby the system cannot make effective recommendations at the beginning of its operation [4, 5, 7]. The “cold start” element within user based approaches refers to the challenge in a-priori knowing what this user, or similar users, will do in new or in the early stages of a given system. It can take weeks, or even months until enough data is collected on new items to attempt a collaborative solution. Even within item based approaches, it is not necessary clear which characteristics should be used to find similar items without any a-priori knowledge. This problem is very significant for MySupermarket as new products are constantly being added to the system and there is no clear connection between the new item and others in the database. Thus, alternative recommendation approaches are necessary.

A third, less popular approach, involves *knowledge based* recommendation [3, 5] which uses some preset rules for generating recommendations. The advantage of this approach is a complete solution to the cold-start problem – accurate recommendations can be immediately generated. The major disadvantage to this approach is the steep overhead involved with the knowledge engineering. MySupermarket currently employs 9 knowledge experts who create rules for generating recommendations for new products. Not only are these rules expensive to generate, but they are not necessarily accurate. The goal of this paper is to describe an approach that uses a knowledge based approach for the early stages of the system, but also create recommender agents that can autonomously update these initial recommendations based on both item based and collaborative approaches.

To the best of our knowledge, this paper represents the first of its kind – a knowledge based approach with item based and collaborative elements to update the original recommendations. Many hybrid recommendation models have been previously suggested with combinations of these approaches and surveys of these models have been previously published [5, 4, 1]. These algorithms often combine the two popular families of recommendation algorithms – collaborative and item-based approaches [1, 8]. Closest to our approach are the Libra [9] and MovieLens [11] systems. However, both of these systems augment collaborative systems with content based approaches. However, many other hybrid combinations are possible, with previous work described a theoretical number of 53 possible different types of hybrid systems [4]. The same article also points out that most theoretical combinations have not been studied or implemented, and particularly singles out directions involving hybrid systems with knowledge based components should be further explored. Particularly, our system goes one step further from previous hybrids, by also integrating expert knowledge along with a more classic content based – collaborative hybrid. We now detail the exact algorithms used by the system, and how the expert’s recommendations are augmented by the item based and collaborative elements.

3 Using MySupermarket’s Expert Data

As most recommendation systems are based on collaborative or item based data that can be cheaply obtained and analyzed [3, 5], it may seem strange that MySupermarket bases

its system on a costly team of experts. In this section, we describe the motivation behind MySupermarket's business decision to use this approach, as well how the company uses this data in creating its recommendation system.

MySupermarket.com's use of experts to create recommendation system is indeed costly. The company employees a team of experts that evaluate thousands of products that are sold through the website, and create an expert measure which they call a *similarity score* which compares all products to each other. To slightly simplify the process, these experts defined "Product Families" of similar products such as types of wines, dairy product, diapers, etc., and only consider creating scores for all products within all given product families. Nonetheless, this process is expensive, as the company employees a team of 9 experts who on average study 100 products a day checking and updating products' similarity rating. The current trigger for this analysis is when new products are added for sale by MySupermarket, thus requiring the experts to reconsider how these new products are comparable to existing ones.

With the growth of automated recommendation systems, one might think that there is no longer a need for this costly knowledge engineering process and these experts should be replaced by automated recommendation agents. However, MySupermarket's use of these expert's knowledge goes well beyond its application for helping recommend products to end users, or its Business to Consumer (B2C) e-commerce website. In addition, these experts' knowledge forms the foundation for a second Business to Business (B2B) application, called MySupermarket insights that provides information about trends and possible strategic growth opportunities related to products supermarkets stock. While our focus is on how the recommendations from the first system can be improved, it should be noted that the second types of recommendations for businesses are no less important to the business strategy of the company and cannot be replaced by known recommendation algorithms. This is because the B2C application has already been functioning for several years and has now created enough historical data to overcome the classic cold start problem in new recommendation systems [4, 5]. However, the B2B application has far less historical data and the experts' knowledge is not easily encodable. For example, these experts maintain a blog about product trends and prices and thus cannot be replaced with automated agents. More about the B2B application, and the recommendations it provides can be found at the company's website at: <http://www.mysupermarket-insights.co.uk/Marketing/Services.aspx>.

In creating the B2C application, the expert's knowledge is central towards deciding what recommendations are presented to the user, and in generating what the company calls "swap recommendations". While shopping, items can be presented to the user that may be of interest, such as items that may save the user money by purchasing them in larger bulk, or alternative products that should be considered, especially when these items are discounted due to sales promotions or are a comparable generic alternative. Furthermore, these recommendations are especially important when the item they wished to buy is not in stock.

The expert's knowledge is then used in conjunction with item based data to create recommendations. Similar to item based recommendation systems, swap recommendations are generated by constructing a similarity vector between the desired product and characteristics of all other products within the company [1, 4, 7, 10]. However, non-

hybrid item based recommenders are based on generic item data, which for this domain are likely to include characteristics like the product family, its quantity, price, weight, and color. In contrast, MySupermarket's hybrid system includes one new characteristic, the expert's similarity measure, and explicitly gives this item with very high weight in generating the vector to decide what products to recommend. Additionally, as opposed to classic item based methods that use machine learning techniques to decide how to weigh each characteristic within the vector, MySupermarket currently uses a hard-coded proprietary weight function between these items. For example, this weight system presents up to 5 recommendation if it finds items that are comparable based on these hard-coded weights taking into account all item's characteristics. In addition, MySupermarket also leaves one field, the last recommendation, where recommendations are based only one characteristic, price alone. Here, the system always presents an alternative if a cheaper generic substitute exists in the product database even if it is not deemed as similar by the other characteristics.

To better understand the system, please see Figure 1 depicting a screen shot from the company's website. Note that in the screenshot the user is given up to four swap recommendations by the system. Only items that are deemed worthy based on this weight function are presented to the user, and thus the full maximal number of 6 recommendations were not presented here. Please note in the first row of Figure 1 that the user is encouraged to consider buying similar diapers in bulk, with the first choice being cheaper than the second, but both being the same brand as the original product, and only then is the user presented a third choice that is a different generic brand, yet far cheaper. In the second row, the user is informed that there is a buy one get one free sale on the item they selected, and she can receive a second product for no additional price. Here no additional products are presented, as the expert's hard coded threshold decides no other products are sufficiently similar given the price differences. Similarly, in the third row, the user is informed there is a sale and she could save money per item if she chooses to buy 2 products instead of one, but no other products are given from different brand. In the last row, the user is again encouraged to consider a sale item or a generic substitute for the selected item.


4 Creating a New Type of Hybrid System

One important question MySupermarket must address is how good are the system's recommendations, and if they are not always effective, how could they be improved? Intuitively, it seems unlikely that the system of static weights described above will always be accurate, especially as the items in the product database are constantly in flux, as sales and changes in stock are frequent. Thus, these static weights do not necessarily have the ability to deal with these dynamics. Furthermore, the need to constantly update these weights is costly. Clearly some mechanism is needed to autonomously update the system.

Towards building a more effective system, we believe a new type of hybrid model is needed, as presented in this section. The basis of this hybrid is the above knowledge

mySupermarket Home - mySupermarket Wine - Register - Sign in

Welcome, (Sign in) £17.58 (Tesco)
















Price Checker
Swap 4 items and save £10.19

CONTINUE

Your selected items >>>>

Smart choices what's this?

Save even more! what's this?

<p>1</p> <div style="text-align: center;">  <p>5 for £10.48 1 (£12.99)</p> </div> <p>£12.99 any 2 FOR £18.00 (19.1p / 13.2p / Nappy) Pampers Simply Dry Size 5 Junior 11-25kg (68)</p> <p style="text-align: center;">362 cal / 100g</p>	<div style="text-align: center;">  <p>2 for £10.00 - Save £6.81</p> </div> <p>£7.29 £5.00 (11.4p / Nappy) Pampers Simply Dry Size 5 Junior 11-25kg (44)</p> <p style="text-align: center;">362 cal / 100g</p>	<div style="text-align: center;">  <p>2 for £18.00 - Save £7.98</p> </div> <p>£12.99 any 2 FOR £18.00 (19.1p / 13.2p / Nappy) Pampers Simply Dry Size 5 Junior 11-25kg (68)</p> <p style="text-align: center;">362 cal / 100g</p>	<div style="text-align: center;">  <p>3 for £10.05 - Save £2.56</p> </div> <p>£3.35 (15.2p / Nappy) Tesco Baby Essentials Size 5 Junior 11-25kg (22)</p> <p style="text-align: center;">231 cal / 100g</p>
<p>2</p> <div style="text-align: center;">  <p>1 (£1.30)</p> </div> <p>£1.30 any 2 FOR 1 (£1.30 / 65p / Cake) McVitie's Jamaica Ginger Cake</p> <p style="text-align: center;">72 cal / 100g</p>	<div style="text-align: center;">  <p>2 for £1.30 - Save £1.30</p> </div> <p>£1.30 any 2 FOR 1 (£1.30 / 65p / Cake) McVitie's Jamaica Ginger Cake</p> <p style="text-align: center;">72 cal / 100g</p>	<p style="text-align: center;">91 cal / 100g</p>	
<p>3</p> <div style="text-align: center;">  <p>1 (£2.14)</p> </div> <p>£2.14 any 2 FOR £3.00 (85.6p / 60p / 100g) John West Tuna Light Lunch Tomato Salsa (250g)</p> <p style="text-align: center;">238 cal / 100g</p>	<div style="text-align: center;">  <p>2 for £3.00 - Save £1.28</p> </div> <p>£2.14 any 2 FOR £3.00 (85.6p / 60p / 100g) John West Tuna Light Lunch Tomato Salsa (250g)</p> <p style="text-align: center;">238 cal / 100g</p>	<div style="text-align: center;">  <p>2 for £3.00 - Save £1.11</p> </div> <p>£2.14 any 2 FOR £3.00 (89.2p / 62.5p / 100g) John West Tuna Light Lunch Mediterranean (240g)</p> <p style="text-align: center;">238 cal / 100g</p>	
<p>4</p> <div style="text-align: center;">  <p>1 (£1.15)</p> </div> <p>£1.15 any 2 FOR £1.50 (14.4p / 9.4p / 100g) Kingsmill Great Everyday Soft White Thick Loaf (800g)</p> <p style="text-align: center;">238 cal / 100g</p>	<div style="text-align: center;">  <p>2 for £1.50 - Save 80p</p> </div> <p>£1.15 any 2 FOR £1.50 (14.4p / 9.4p / 100g) Kingsmill Great Everyday Soft White Thick Loaf (800g)</p> <p style="text-align: center;">238 cal / 100g</p>	<div style="text-align: center;">  <p>2 for £1.50 - Save 80p</p> </div> <p>£1.20 any 2 FOR £1.50 (15p / 9.4p / 100g) Kingsmill Great Everyday Medium Sliced Soft White Bread (800g)</p> <p style="text-align: center;">238 cal / 100g</p>	<div style="text-align: center;">  <p>1 for 74p - Save 41p</p> </div> <p>74p (9.3p / 100g) Tesco Thick Sliced White Loaf (800g)</p> <p style="text-align: center;">231 cal / 100g</p>

£
CONTINUE

Fig. 1. A Sample Webpage from MySupermarket's Website

based system, which is useful for providing initial recommendations and is critical for other MySupermarket applications. However, once a sufficient history is stored through system use, item based and collaborative components can be potentially useful in improving the system. However, one key question that must be addressed is when and how can this data be useful in improving the system. Thus, care must be taken to properly evaluate the usefulness of this added information, as we now detail.

4.1 A High Level System Overview

We propose constructing a three pronged hybrid that is knowledge based, but uses item based and collaborative elements. A high level overview of our solution is shown in Figure 3. As per MySupermarket's business model, the Knowledge Based component is at the core of the system and is shown at the top left corner of the diagram. As people begin using the system, historical data is accumulated and this data is sent as input into item based and collaborative components. If this data is found to be useful, a hybrid model is formed where these models can be used in several ways: First, and on the most basic level, assuming the expert's knowledge is not equivalent to these models, we can manually query the expert for input. It may be the expert will then wish to manually revise or accept the values automatically generated by these components. However, as we have begun to find, the experts are willing to forgo this step, thus automatically accepting the autonomously generated agent changes. The outcome is a revised hybrid system, that began exclusively as being knowledge based, but has accepted many key components from the item based and collaborative algorithms.

To better understand the process by which the knowledge based recommender is modified, please refer to Algorithm 1. As lines 1 and 2 state, initially the experts must manually evaluate every item within the system, assigning a similarity value for every product versus all other products. This similarity values is then evaluated in conjunction with all other item attributes in a hard-coded formula to produce the system's initial recommendations. However, as the system is used, some critical size of product history is likely to become available for this product (line 5), to reevaluate these initial knowledge based recommendations. Assuming this is the case, we currently perform three checks. First, in line 6, we evaluate the overall effectiveness for the recommendation output of this product. We found that for many products the users were willing to accept the system's recommendations, and for others users almost never accepted the system's recommendation. Currently, we simply flag those products with a very low user acceptance of the system's recommendations (line 6) and present these results to the experts for consideration. However, our goal is to automate any such evaluations through allowing the recommender agents to autonomously change the system. To accomplish this, we use verify and change the system through item-based and collaborative data when available. In line 8, agents automatically evaluate the effectiveness of the expert's hard-coded initial weights through machine learning techniques, e.g. decision trees, as described in the next subsection. Assuming this item-based model is not built around the expert's information (line 9), the system can either prompt the expert to accept the item

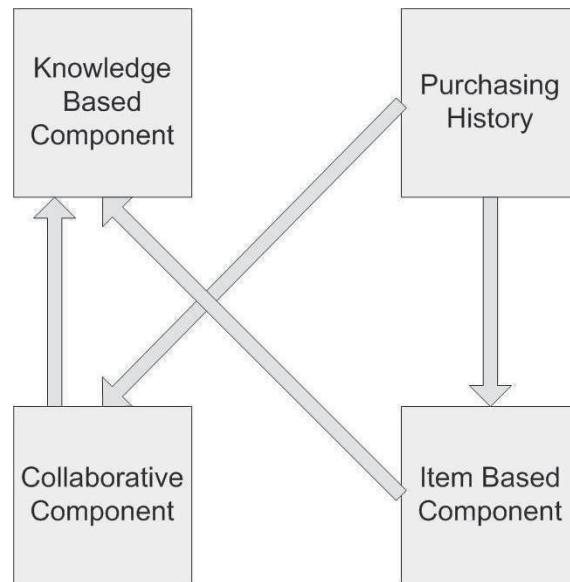


Fig. 2. An architecture of a Hybrid Recommender System that is based on expert knowledge, but also revises the system with item-based and collaborative components.

based recommendations or as we have begun to allow, autonomously update the system (line 10). Furthermore, the recommender agent checks the initial expert's recommendations against acquired collaborative data (line 11). Assuming these weights are not equal (line 12), we again either prompt the user to accept the changes or automatically update the system.

Algorithm 1 *The major steps for dynamically updating / changing the recommendation system*

```

01 for Every product in System do
02   Create initial recommendations based on Expert's Knowledge
03 while the System is in use do
04   for Every product in System do
05     if data history exists for this product then
06       if User acceptance for product < threshold then
07         Flag product in system
08         Build Item Based Model with Decision trees
09       if Expert's Information not the root of the decision tree then
10         Present findings to Expert / Accept Item Based Recommendations
11       if Hybrid-Item weight  $\neq$  Collaborative Values then
12         Present findings to Expert / Accept Collaborative Values

```

As Algorithm 1 indicates, the recommender system is one in flux, beginning exclusively based on expert knowledge, but allows agents to autonomously update the initial

system. However, in doing so, several challenges exist with implementing this algorithm, which are addressed in the following subsection. All three system checks of the expert's initial recommendations (lines 6 – 12) are built around the assumption that the recommender system can be objectively be evaluated. However, as we present in the next Section (4.2), evaluating recommender systems is far from trivial, especially if a controlled dataset cannot be formed. Second, we present a novel approach where agents can check the expert's recommendations using item based information. This again is not simple, and our approach for doing so is presented in Section 4.3. Finally, the use of collaborative data is again non-trivial, and our approach for doing so is presented in Section 4.4.

4.2 Evaluating the Overall System

MySupermarket's B2B and B2C applications are both built on their experts' knowledge. Thus, the key question about the accuracy of the expert's knowledge is not limited to the recommendations for their e-commerce website, but also for their B2B application as well. In general, many metrics have been proposed to date to evaluate the effectiveness of recommendation systems [7]. For example, one popular choice, used in the Netflix competition [2] is to use the root mean error level of prediction between a set of previously tagged known ratings that people provide, and a set of automatically generated recommendations by the system. However, this possibility is not available to us, as we have no previously tagged data to use as a baseline. Instead, we use the bottom line user satisfaction measure most intuitive to use in commercial systems [7].

We propose that two types of bottom line measures are useful in evaluating the expert's knowledge of this system. The first, and possibly more intuitive measure is to measure the number of purchases made because of the recommended product swaps. As the company has logged all transactions to its website over the past 5 years, extensive historical data is available to allow for this analysis. A second complementary measure searches for statistical correlation between those elements that were swapped in the past (line 5 of Algorithm 5) and the expert's recommendations. Note that the two studies are intrinsically linked: If no swaps are performed, the recommendation system is clearly not producing quality alternatives, and no correlation will be found between people's decisions and their swap purchases. If swaps are frequently performed, the question then becomes, "why"? Are these swaps due to something inherent with these products, or due to the expert's knowledge, both factors, or something else?

We found that the number of swap purchases made varied greatly between different product families. Figure 2 presents a look at 5 different product families and their average number of executed "swaps" or acceptance of the system's recommendation. Note that these 5 product families are a small samples of the 950 product families within the system. However, we did find overall great differences in the acceptance of the system's recommendations across different types of products. Intuitively, such differences may be because people are naturally more picky about accepting certain product substitutions other others. For example, we found that people looking to buy a certain type of

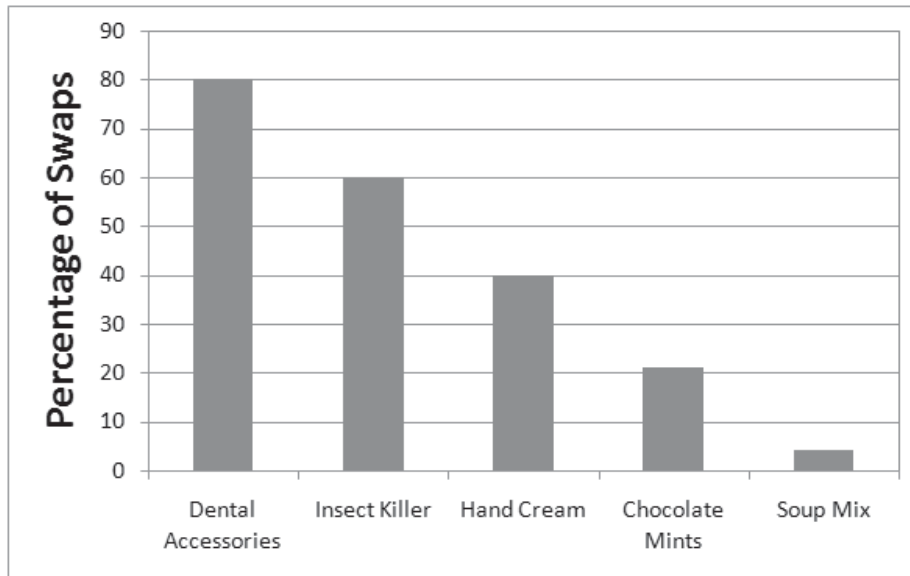


Fig. 3. Five Different Product Families and their Average Number of Accepted Recommendations

dental accessories (e.g. dental floss) were most likely to accept the system's recommendation and chose an alternate product approximately 80% of the time. However, people who were looking to buy a certain type of insecticide were only nearly 60% likely to accept the system's recommendation, and people looking for hand cream were accepted the system's recommendation about 40% of the time. The percentage of times users accepted certain recommendations were extremely low, such as slightly more than 20% for chocolate mints, and less than even 5% for soup mixes. Overall we found that these examples represent a wide range of acceptance levels, and that people accepted the system's recommendations approximately 35% of the time. However, is this level of success due to some inherent pickiness of users about some types of products versus others, or is this the truly optimal state? If it is not the optimal state, what changes would be necessary to further improve the system's performance?

At present, MySupermarket uses this swap analysis to create a report to the experts. The experts are then asked to manually analyze the data to question if their knowledge is in fact effective in generating more sale. For example, we may present the system's 5% success in generating swaps for soup mixes and ask the expert to manually change its recommendation scheme for the products in this group. However, the company's vision involves using autonomous agents to automatically update these expert's values, as described in the following sections.

4.3 Evaluating the System with Item Data

Our first goal was to verify and update the expert's similarity measure through using machine learning techniques to check the predictive ability of the expert's informa-

tion. To do so, we use the well recognized Weka [12] package to create a predictive model regarding when people purchased a product from the among the system's recommendations. Realistically, some complex relationship likely exists between the type of product, the quality of the expert's information, the possible savings to the user, and other factors in determining if a swap purchase is made. For example, this analysis may find that price is used for some categories, other products are only swapped when the expert's similarity measure is less than a certain amount, and certain products are never swapped.

The use of machine learning techniques to validate the recommendation model is a twist from the classic use of these algorithms within item based recommenders. In classic item-based classification, a collection of all item characteristics are used in conjunction with historical data about purchases to create a learned model that correlates between the two [10]. This type of learning can use any machine learning algorithm, including Bayes, decision trees, and nearest neighbor methods to accurately find a correlation between items, their characteristics, and historical data. No a-priori assumption is made as to which characteristics will make the best model – in fact the purpose of the model is to find these characteristics. In contrast, our goal is exactly the opposite. The expert has already decided and hard-coded her own similarity measure as being most important, and fixed the relative value of all other item characteristics. In the best case scenario, the expert has discovered certain domain specific knowledge, encapsulated in its similarity measure, allowing it to surpass the recommendations of a pure item based system. Alternatively, the item based rules may approximate the expert based knowledge, and comparing the derived rules will allow us to confirm the accuracy of the expert knowledge. However, the pure item based system might be more accurate, allowing us to pinpoint for exactly which items the expert's knowledge is less accurate.

We chose to evaluate the expert's knowledge through creating a model based on decision trees. The advantage towards using trees versus any other model is that Weka [12] not only creates a machine learning model, but also outputs the exact rules used in this model. Assuming the expert's knowledge is critical to the system, one would expect to find the expert's similarity measure to be the key rule, or at the root of the decision tree. If the expert's knowledge is not effective, one would expect it to either not appear in the tree, or be limited to only very specific instances.

In creating these decision trees, we used as input the history of people's swap purchases for given product families, and entered all items' data into Weka [12]. The item's input data included information the expert's similarity measure, the projected saving by choosing the new item, as well as items characteristics not currently given significant weight by the experts, such as the serial number of the product and the serial number of the proposed product. We recognize that it is quite possible that items the overlooked, say the serial number of the proposed product, may produce recommendations that the experts overlooked.

For many product families, we were able to confirm the importance of the expert's knowledge, while for other products the expert's knowledge seemed much less important. For example, Weka's decision tree for purchases made for squash had at the root of the tree: $\text{similarity} \leq 1.25$, or if the similarity measure is less than 1.25, then people are likely to buy in certain conditions. In other product families, such as for milk

products, the similarity function was of secondary importance to the difference in cost between products. Here we found the rule: If the $\text{AlternativePrice} < 0.85$ and $\text{similarity} < 1.1$, then given certain other conditions the person will purchase the product. However, for other product families similarity had seemingly no significance. For example, for toilet paper the root rule was if the $\text{OriginalPricePerUnit} \leq 0.35$ and the $\text{AlternativePricePerUnit} \leq 0.28$, then a person will buy given other conditions. Thus, we found that using decision trees were useful in automatically generating where the expert's knowledge was most useful.

Note that as per line 9 of Algorithm 1, two possibilities exist when decision trees found that the expert's similarity measure was not the most important item characteristic. Until recently, this information was presented to the expert, who could then decide if she would like to revise the values, or accept the decision tree's rules instead. However, we have begun a pilot whereby the agent autonomously updates the expert's recommendation, especially for products where the expert's recommendations yielded a low recommendation (e.g. set the threshold of line 6 of Algorithm 1 to 10%).

4.4 Evaluating the System with Collaborative Data

We also use historical data to create a collaborative model to augment the expert's recommendations. The above machine learning approach to validate the expert's similarity measure can validate the importance of this item to the recommender agent for how the average, or typical user, behaved. Furthermore, the weights set by the expert, and even by the hybrid knowledge-item based system, are still uniform across all users. However, this approach does not validate how a specific user behaved, and if this model is appropriate for a specific user. For example, the experts may have hard-coded the system to only present alternatives where a similarity value of 1.0 or less is found. However, it may be found that certain users are willing to buy items that are even less similar (e.g. values of greater than 1.0) and some are more discriminating and only purchase items that are far more similar (say similarity 0.5 or less). Thus, the above approach can only verify that user's in general are willing to make purchases based on the expert's measures, it cannot predict if a specific user deviates from this assumption.

Note that the difference of the behavior of a general user and the behavior of a specific user is the inherent difference between item-based and collaborative recommendation systems. As our goal is to customize the system's recommendations as much as possible, we present a heuristic approach where the hybrid knowledge-item based agent's recommendations are further customized based on that specific user's history.

In general, we found that users generally decide to purchase a product based on the expert's similarity measure and the potential cost savings of the new item. However, while we found that these two attributes were important across all users, and thus formed an effective hybrid item-knowledge based system, the actually savings and similarity measures used by a specific user could differ greatly. To address this issue, we found that an heuristic approach, where the similarity and savings measures were tuned based on a specific user's past activity for a given product, was highly effective in improving the system's recommendations. This led to an effective automatic tuning of these parameters, increasing the companies sales through customers' swaps.

In general, it is important to stress that the company’s experts were initially extremely hesitate to forgo their initial values in favor of these found by the item based and collaborative elements as described in the paper. This issue is further complicated by the fact that the system lacks any proper evaluating dataset, and thus it was extremely difficult to convince the experts of the importance of the agent’s recommendations. We overcame this obstacle by first revising the systems only for those products where the initial success of the expert’s system was extremely low (see line 6 of Algorithm 1). This work is ongoing, and will take nearly a year before we can quantify where this approach was successful. However, the generality of this approach leads and our initial feedback from the company’s experts have led us to be confident about its importance.

5 Conclusions and Future Work

In this paper we introduced a novel hybrid approach to combine a knowledge based recommender system with item based and collaborative filtering elements. The system’s recommender agent begins with a system exclusively based on the expert’s knowledge, thus avoiding the classic cold start problem. However, as the system is used, a progressively larger history of user transactions are recorded. The system then uses this information to create hybrid models with item and collaborative items. An item based model is used to validate or even replace the user’s knowledge. We describe using a novel variation of machine learning techniques to create a classic item based model can be used to validate the expert’s knowledge. When the item based model finds the expert’s knowledge is at the root of the item based model, the expert’s knowledge is accepted. When it is found to not be a critical item in the model, the system can prompt the expert to update item data, or automatically replace and update the user’s knowledge. Additionally, if the expert’s knowledge is validated by the item based model, collaborative models are useful for further improving the system’s recommendations by automatically tweaking the system’s item’s parameters based on a specific user’s purchases. We present the system’s prototype implementation and initial results demonstrating the importance and success of this approach.

Several related problems are worthy of future consideration. One key hurdle we needed to overcome was convincing the data experts that the agent’s item and collaborative recommendations should replace or augment their own. We hope to further study at what point can one assume the agent’s recommendations are definitive, and how to convince the experts of this. Achieving this goals would significantly aid us in the goal of fully automating system revisions. Additionally, we hope to further address how the system’s evaluation can be better automated without explicitly labeled data as is done in many classic recommendation system’s, such as the Netflix challenge [2]. We believe the approach we present, of using machine learning techniques to create an item based approach for evaluation, can be further generalized to address this point. The importance of hybrid systems such as the knowledge, item and collaborative system we present, are likely to be of significance to other areas and fields as well. It is likely that use of expert information can help avoid the “cold start” problem in other problems as well. Our model, where collaborative and item based information are later used, are

likely to be equally useful for these problems as well. We hope to study what modifications to our approach are necessary, if any, in addressing new problems.

References

1. Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 17(6):734–749, 2005.
2. James Bennett and Stan Lanning. The netflix prize. In *In KDD Cup and Workshop in conjunction with KDD*, pages 3–6, 2007.
3. Robin Burke. Knowledge-based Recommender Systems. In *Encyclopedia of Library and Information Systems*, volume 69, 2000.
4. Robin D. Burke. Hybrid recommender systems: Survey and experiments. *User Model and User-Adapted Interaction*, 12(4):331–370, 2002.
5. Robin D. Burke. Hybrid web recommender systems. In *The Adaptive Web*, pages 377–408, 2007.
6. David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35:61–70, December 1992.
7. Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22:5–53, January 2004.
8. Prem Melville, Raymod J. Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Eighteenth national conference on Artificial intelligence*, pages 187–192. American Association for Artificial Intelligence, 2002.
9. Prem Melville, Raymod J. Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Eighteenth national conference on Artificial intelligence*, pages 187–192, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.
10. Michael Pazzani and Daniel Billsus. Content-Based Recommendation Systems. pages 325–341. 2007.
11. Badrul M. Sarwar, Joseph A. Konstan, Al Borchers, Jon Herlocker, Brad Miller, and John Riedl. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work, CSCW '98*, pages 345–354, 1998.
12. Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Morgan Kaufmann, June 2005.

Modeling and Evaluating Human-Like Behavior in Autonomous Agents playing the Social Ultimatum Game

Yu-Han Chang, Tomer Levinboim., and Rajiv Maheswaran

Information Sciences Institute, University of Southern California
Marina Del Rey, CA 90292, USA

Abstract. We address the challenges of evaluating the fidelity of autonomous agents that are attempting to replicate human behaviors. This is a fundamental issue in the emerging intersection of artificial intelligence and social science motivated by problems such as training in virtual environments and large-scale social simulation. Our specific interest focuses on emulating human strategic behavior over time. We introduce and investigate the Social Ultimatum Game and discuss the efficacy of a set of metrics in comparing various autonomous agents to human behavior collected from experiments.

1 Introduction

We address the challenge of building autonomous agents that exhibit human-like behavior in economic settings and more importantly, developing techniques for evaluating their efficacy. In particular, we are interested in multi-agent domains where humans make sequential decisions over time. In many settings, agent “goodness” is typically evaluated relative to optimal behavior, using a metric like expected reward. However, realistic human behavior is often not optimal, and in many of the domains of interest, the notion of optimality is ill-defined.

Optimality of one agent in a multi-agent domain is dependent on the other agents. If a machine’s assumptions about the other agents is incorrect, then its behavior, even if optimal given those assumptions, could be wildly different from normal human behavior. We will see an example of this shortly, in a variant of the classic Ultimatum game. Since the validity of these assumptions is an essential part of what must be evaluated, optimality based on the assumptions is not a good metric for realism. We need a different approach.

Human data in multi-agent domains is getting easier to collect, given the current state of access to the Internet and online interaction. Thus, we can obtain baseline collections of behavior trajectories that describe human play. The challenge is to find a way to compare collections of traces produced by autonomous agents with this existing baseline, in order to determine which agents exhibit the most realistic behavior.

In this paper, we investigate these issues in the context of the Social Ultimatum Game (SUG). SUG is a multi-agent multi-round extension of the Ultimatum Game [5], which has been a frequently studied game over the last three decades as a prominent example of how human behavior deviates from game-theoretic predictions that use the “rational actor” model. Data gathered from people playing SUG was used to create

various classes of autonomous agents that modeled the behaviors of the individual human players. We then created traces from games with autonomous agents emulating the games that the humans played. We develop several metrics to compare the collections of traces gathered from games played by humans and games played by the autonomous agents. From this analysis, it becomes clear that human behavior contains unique temporal patterns that are not captured by the simpler metrics. In SUG, this is revealed in the likelihood of reciprocity as a function of the history of reciprocity. The key implication is that it is critical to retain the temporal elements when developing metrics to evaluate the efficacy of autonomous agents for replicating human strategic behavior in dynamic settings.

2 The Social Ultimatum Game

To ground our subsequent discussion, we begin by introducing the Social Ultimatum Game. The classical Ultimatum Game, is a two-player game where P_1 proposes a split of an endowment $e \in \mathbb{N}$ to P_2 who would receive $q \in \{0, \delta, 2\delta, \dots, e - \delta, e\}$ for $\delta \in \mathbb{N}$. If P_2 accepts, P_2 receives q and P_1 receives $e - q$. If P_2 rejects, neither player receives anything. The subgame-perfect Nash or Stackelberg equilibrium has P_1 offering $q = \delta$ (i.e., the minimum possible offer), and P_2 accepting, because a “rational” P_2 should accept any $q > 0$, and P_1 knows this. Yet, humans make offers that exceed δ , make “fair” offers of $e/2$, and reject offers greater than the minimum.

To represent the characteristics that people operate in societies of multiple agents and repeated interactions, we introduce the Social Ultimatum Game. Players, denoted $\{P_1, P_2, \dots, P_N\}$, play $K \geq 2$ rounds, where $N \geq 3$. In each round k , every player P_m chooses a recipient R_m^k and makes them an offer $q_{m,n}^k$ (where $n = R_m^k$). Each recipient P_n then considers the offers they received and makes a decision $d_{m,n}^k \in \{0, 1\}$ for each offer $q_{m,n}^k$ to accept (1) or reject (0) it. If the offer is accepted by P_m , P_m receives $e - q_{m,n}^k$ and P_n receives $q_{m,n}^k$, where e is the endowment to be shared. If an offer is rejected by P_n , then both players receive nothing for that particular offer in round k . Thus, P_m 's reward in round k is the sum of the offers they accept (if any are made to them) and their portion of the proposal they make, if accepted:

$$r_m^k = (e - q_{m,n}^k)d_{m,n}^k + \sum_{j=1 \dots N, j \neq m} q_{j,m}^k d_{j,m}^k \quad (1)$$

The total rewards for P_m over the game is the sum of per-round winnings, $r_m = \sum_{k=1}^K r_m^k$. A game trajectory for P_m is a time-series of proposed offers, $O_m^k = (R_m^k, q_{m,n}^k, d_{m,n}^k)$ and received offers, $O_{n,m}^k = (R_n^k, q_{n,m}^k, d_{n,m}^k)$. At time k , the trajectory for P_m is $T_m^k = (O_m^k, \{O_{n,m}^k\}_n, O_m^{k-1}, \{O_{n,m}^{k-1}\}_n, \dots, O_m^1, \{O_{n,m}^1\}_n)$. Assuming no public information about other players' trajectories, T_m^k includes all the observable state information available to P_m at the end of round k .

3 Metrics

Let C_m be the collection of trajectories P_m produces by taking part in a set of Social Ultimatum Games. In other domains, these traces could represent other interactions. Our goal is to evaluate the resemblance of a set of human trace data C to other sets of traces \tilde{C} , namely those of autonomous agents. We need a metric that compares sets of multi-dimensional time series: $d(C, \tilde{C})$. Standard time-series metrics such as Euclidean or absolute distance, edit distance, and dynamic time warping [9] are not appropriate in this type of domain.

One challenge arises because we are interested in the underlying behavior that creates the trajectories rather than superficial differences in the trajectories themselves. If we can collapse a collection of traces C to a single probability distribution Q , by aggregating over time, then we can define a *time-collapsed* metric,

$$d(C, \tilde{C}) = KL(Q||\tilde{Q}) + KL(\tilde{Q}||Q) \quad (2)$$

where KL denotes the Kullback-Leibler divergence. The sum enforces symmetry and nonnegativity. Time-collapsed metrics for SUG include:

- **Offer Distribution.** Let Q^O be the distribution of offer values $\{q_{m,n}^k\}$ observed over all traces and all players.
- **Target-Recipient Distribution.** Let Q^R denote the likelihood that a player will make an offer to the k^{th} most likely recipient of an offer. This likelihood is non-increasing in k . In a 5-person game, a single player may have an target-recipient distribution that looks like $\{0.7, 0.1, 0.1, 0.1\}$ which indicates that they made offers to their most-targeted partner 7 times more often than their second-highest-targeted partner. We can produce Q^R by averaging over all games to characterize a player and further average over all players to characterize a population.
- **Rejection Probabilities.** For each offer value q , we have a Bernoulli distribution Q^{B_q} that captures the likelihood of rejection by averaging across all players, games and rounds in a collection of traces. We then define a metric:

$$d^B(C, \tilde{C}) = \sum_{q=0}^{10} KL(Q^{B_q}||\tilde{Q}^{B_q}) + KL(\tilde{Q}^{B_q}||Q^{B_q}).$$

We can also define *time-dependent* metrics that acknowledge that actions can depend on observations of previous time periods. One prominent human manifestation of this characteristic is reciprocity. We define two time-dependent metrics based on reciprocity:

- **Immediate Reciprocity** When a player receives an acceptable offer from someone, they may be more inclined to reciprocate and propose an offer in return in the next round. We can quantify this $p(R_m^{k+1} = n | R_n^k = m)$ across all players and games in a collection of traces. This probability defines a Bernoulli distribution Q^Y from which we can define a metric d^Y as before.

- **Reciprocity Chains** Taking the idea of reciprocity over time further, we can calculate the probability that an offer will be reciprocated, given that a chain of reciprocity has already occurred. For example, for chains of length $c = 2$, we $p(R_m^{k+1} = n | R_n^k = m, R_m^{k-1} = n)$; for $c = 3$, we calculate $p(R_m^{k+1} = n | R_n^k = m, R_m^{k-1} = n, R_n^{k-2} = m)$. As before, these probabilities can be used to define a Bernoulli distribution Q^{Y_c} for each length c . Then, for some L , we define

$$d_L^Y(C, \tilde{C}) = \sum_{c=1}^L KL(Q^{Y_c} || \tilde{Q}^{Y_c}) + KL(\tilde{Q}^{Y_c} || Q^{Y_c}).$$

We expect that the longer a pair of players reciprocate, the higher the likelihood that they will continue doing so. The probabilities of how likely humans are to reciprocate can be obtained from the experimental data.

4 Autonomous Agents

In this section, we describe various agent models of behavior. We first apply traditional game-theoretic analysis to the Social Ultimatum Game to derive the “optimal” behavior under rational actor assumptions. We then describe two distribution-based agents that do not model other agents but are capable of incorporating human behavior data. Finally, we describe an adaptive agent that incorporates some aspects of human behavior such as fairness and reciprocity.

4.1 Game-Theoretic Agents

Let strategies be characterized by the statistics that they produce in steady-state: the distribution of offers made by each player, where $p_m^g(n, q)$ denotes the likelihood that P_m will give an offer of q to P_n , and the distribution of offers accepted by each player, where $p_m^a(n, q)$ denotes the likelihood that P_m will accept an offer of q from P_n . Then, the expected reward for P_m per round in steady-state is $r_m =$

$$\sum_{n,q} qp_n^g(m, q)p_m^a(n, q) + \sum_{n,q} (e - q)p_m^g(n, q)p_n^a(m, q) \quad (3)$$

where $\sum_{n,q} p_m^g(n, q) = 1, \forall m$, as the total outgoing offers must total one offer per round, and the acceptance likelihoods are $p_m^a(n, q) \in [0, 1], \forall m, n, q$. A player maximizing these rewards will modify their offer likelihoods $\{p_m^g(n, q)\}$ and acceptance likelihoods $\{p_m^a(n, q)\}$, given those of other players. A player can create the desired statistics by playing a stationary mixed strategy with the desired likelihoods. To optimize the offer likelihoods, P_m sets

$$p_m^g(n, q) > 0, \forall n \in \mathcal{N}^g \subset \arg \max_n \max_q (e - q)p_n^a(m, q)$$

such that $\sum_{n,q} p_m^g(n, q) = 1$, and $p_m^g(n, q) = 0$, otherwise. Thus, in equilibrium, P_m will make offers to those agents whose acceptance likelihoods yield the highest expected payoff.

Proposition 1. *In the Social Ultimatum Game, accepting all offers is not a dominant strategy.*

Proposition 2. *In the Social Ultimatum Game, Nash equilibrium outcomes only happen when players employ strategies of the form “greedy” strategies, where*

$$p_m^g(n, q) = 0, \forall q > \delta, m, n, \quad p_m^a(n, \delta) = 1, \forall m, n, \quad (4)$$

i.e., “greedy” strategies where players only make the minimum offers of δ , and all players accept all minimum offers.

Proof details are provided in [2].

4.2 Distribution-Based Agents

One way to create agents that satisfy a set of metrics is to use the metrics to generate the agent behavior. Using only time-collapsed metrics, one could create a distribution-based agent (DBA) as follows. Learn distributions of offer value, target recipient and rejection percentage from human data. Find the appropriate target-recipient distribution based on number of participants and assign agents to each position (i.e., most likely to least likely). In offer phases of each round, choose a target by sampling from the target-recipient distribution and an offer value by sampling from the offer distribution. For received offers, decide via Bernoulli trial based on the rejection percentage for that offer value.

The DBA has no notion of reciprocity. We also investigated a class of distribution-based reciprocal agents (DBRA) which behave like the DBA agents in all aspects other than target selection. If DBRA agents receive an offer it will decide to reciprocate based on a reciprocation percentage that is learned from human data. If multiple offers are received, the target is chosen using a relative likelihood based on the target-recipient distribution. Similarly, if it doesn’t receive any offers, it uses the target-recipient distribution. While the distribution-based agents act on the basis of data of human play, they do not have models of other agents and consequently execute an open-loop static policy. The following model introduces an adaptive model that is not based simply on fitting the metrics.

4.3 Adaptive Agents

In order to create adaptive agent models of human players for the Social Ultimatum Game, we need to incorporate some axioms of human behavior that may be considered “irrational”. The desiderata that we address include assumptions that people will (1) start with some notion of a fair offer, (2) adapt these notions over time at various rates based upon their interactions, (3) have models of other agents, (4) choose the best option while occasionally exploring for better deals. Each player P_m is characterized by three parameters: α_m^0 : P_m ’s initial acceptance threshold, β_m : P_m ’s reactivity and γ_m : P_m ’s exploration likelihood.

The value of $\alpha_m^0 \in [0, e]$ is P_m ’s initial notion of what constitutes a “fair” offer and is used to determine whether an offer to P_m , i.e., $q_{n,m}^k$, is accepted or rejected. The

value of $\beta_m \in [0, 1]$ determines how quickly the player will adapt to information during the game, where zero indicates a player who will not change anything from their initial beliefs and one indicates a player who will solely use the last data point. The value of $\gamma_m \in [0, 1]$ indicates how much a player will deviate from their “best” play in order to discover new opportunities where zero indicates a player who never deviates and one indicates a player who always does.

Each player P_m keeps a model of other players in order to determine which player to make an offer to, and how much that offer should be. The model is composed as follows: $a_{m,n}^k$: P_m 's estimate of P_n 's acceptance threshold; $\bar{a}_{m,n}^k$: Upper bound on $a_{m,n}^k$; and $\underline{a}_{m,n}^k$: Lower bound on $a_{m,n}^k$. Thus, P_m has a collection of models for all other players $\{[\underline{a}_{m,n}^k, a_{m,n}^k, \bar{a}_{m,n}^k]\}_n$ for each round k . The value $a_{m,n}$ is the P_m 's estimate about the value of P_n 's acceptance threshold, while $\underline{a}_{m,n}^k$ and $\bar{a}_{m,n}^k$ represent the interval of uncertainty over which the estimate could exist. For simplicity, we will assume that $\delta = 1$.

Making Offers In each round k , P_m may choose to make the best known offer, denoted \tilde{q}_m^k , or explore to find someone that may accept a lower offer. If there are no gains to be made from exploring, i.e., the best offer is the minimum offer ($\tilde{q}_m^k = \delta = 1$), a player will not explore. However, if there are gains to be made from exploring, with probability γ_m , P_m chooses a target P_n at random and offers them $q_{m,n}^k = \tilde{q}_m^k - 1$. With probability $1 - \gamma_m$, P_m will choose to exploit. The target is chosen from the players who have the lowest value for offers they would accept, and the offer is that value:

$$q_{m,n}^k = \lceil a_{m,n}^k - \epsilon \rceil \text{ where } n \in \arg \min_{\bar{n} \neq m} \lceil a_{m,\bar{n}}^k \rceil \quad (5)$$

The previous equation characterizes an equivalence class of players from which P_m can choose a target agent. The ϵ parameter is used to counter boundary effects in the threshold update, discussed below. The target agent from the equivalence class is chosen using *proportional reciprocity*, by assigning likelihoods to each agent with respect to offers made in some history window.

Accepting Offers For each offer $q_{m,n}^k$, the receiving player P_n has to make a decision $d_{m,n}^k \in \{0, 1\}$ to accept or reject it, based on its threshold:

$$\text{If } q_{m,n}^k \geq \lceil \alpha_m^k - \epsilon \rceil, \text{ then } d_{m,n}^k = 1, \text{ else } d_{m,n}^k = 0 \quad (6)$$

Updating Acceptance Threshold The acceptance threshold is a characterization of what the agent considers a “fair” offer. Once an agent is embedded within a community of players, the agent may change what they consider a “fair” offer based on the received offers. We model this adaption using a convex combination of the current threshold and the offers that are received, with adaptation parameter β_m . Let the set of offers that are received be defined as: $R_m^k = \{q_{i,j}^k : j = m, q_{i,j}^k > 0\}$. If $|R_m^k| \geq 1$, then $\alpha_m^{k+1} =$

$$(1 - \beta_m)^{|R_m^k|} \alpha_m^k + \frac{(1 - ((1 - \beta_m)^{|R_m^k|})}{|R_m^k|} \sum_i q_{i,m}^k \quad (7)$$

If $|R_m^k| = 0$, then $\alpha_m^{k+1} = \alpha_m^k$. Thus, offers higher than your expectation will raise your expectation and offers lower than your expectation will lower your expectation at some rate.

Updating Threshold Estimate Bounds As a player makes an offer $q_{m,n}^k$ and receives feedback on the offer $d_{m,n}^k$, they learn about P_n 's acceptance threshold. Using this information, we can update our bounds for our estimates of their threshold. The details can be found in an extended version of this paper.

Updating Threshold Estimates Once the threshold bounds are updated, we can modify our estimates of the thresholds as follows. If the player accepts the offer, we move the estimate of their threshold closer to the lower bound and if the player rejects the offer, we move our estimate of their threshold closer to the upper bound using a convex combination of the current value and the appropriate bound as follows.

$$d_{m,n}^k = 1 \Rightarrow a_{m,n}^{k+1} = \min\{\beta_m \underline{a}_{m,n}^{k+1} + (1 - \beta_m) a_{m,n}^k, \bar{a}_{m,n}^{k+1}\} \quad (8)$$

$$d_{m,n}^k = 0 \Rightarrow a_{m,n}^{k+1} = \max\{\beta_m \bar{a}_{m,n}^{k+1} + (1 - \beta_m) a_{m,n}^k, \underline{a}_{m,n}^{k+1} + 2\epsilon\} \quad (9)$$

The *min* and *max* operators ensure that we don't make unintuitive offers (such as repeating a just rejected offer), if our adaptation rate is not sufficiently high. The adaptive agent described above fulfills the properties of the desiderata prescribed to generate behavior that is more aligned with our expectations in reality.

5 Experiments

Data was collected from human subjects recruited from undergraduates and staff at the University of Southern California. In each round, every player is given the opportunity to propose a \$10 split with another player of their choosing. Games ranged from 20 to 50 rounds. A conversion rate of 10 ultimatum dollars to 25 U.S. cents was used to pay participants, i.e., \$5 per 20 rounds per player in an egalitarian social-welfare maximizing game. The subjects participated in organized game sessions and a typical subject played three to five games in one session. Between three and seven players participated in each game. During each session, the players interacted with each other exclusively through the game's interface on provided iPads, shown in Figure 1. We have collected data from 27 human subject games thus far. In this paper, we focus on the seven 5-person games in the dataset. By restricting our attention to five-player games, we avoid biases that may be introduced if we attempted to normalize the data from the other games to reflect a five-person composition. Analysis on the games of other sizes yields similar results.

To create the Distribution-Based Agent and Distribution-Based Reciprocal Agent to the collected data, we calculated the appropriate distributions (offer value, rejection

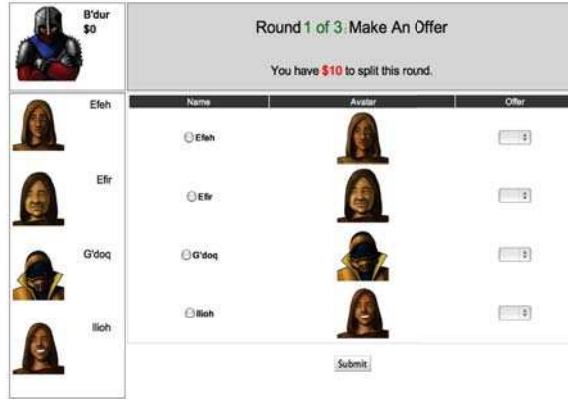


Fig. 1. The Social Ultimatum Game Interface

percentage by value, targeted-recipient), by counting and averaging over all games and all players. For the Adaptive Agents, we analyzed the traces of each game, and estimated game-specific α, β, γ parameters of each of the participating players, as follows. For each player P_m in the game,

- α_m : This is set as the player’s first offer in this game.
- β_m : When a player decreases his offer to a specific player from q_1 to q_2 after K steps (not necessarily consecutive), we find and store the best β value such that K applications of $\beta q_2 + (1 - \beta)q_1$ yields a result less than $\frac{q_1 + q_2}{2}$ (so that the next offer should be closer to q_2 than it is to q_1). We then take β_m to be this stored β value.
- γ_m : This is the likelihood that a player’s offer is less than the minimum known accepted offer, where the minimum accepted offer at a given round k is the minimum offer known to be accepted by any player at time $k - 1$.

Having estimated the population parameters of each game, we then use them as input to create an autonomous agent for each player, and simulate each game ten times to produce ten traces. Within each of these games, each of the five players uses the parameters corresponding to one of the five original human players.

6 Evaluation

These experiments and simulations result in a collection of game traces for each of the five types of agent discussed: Human, Adaptive, DBA, DBRA, and Game-theoretic (GT). Table 1 shows the similarity between the collection of human traces and each of the four collections of autonomous agent traces, according to the metrics discussed earlier.

The DBA and DBRA agents score very well on the three metrics based on offer value, rejection percentage, and target-recipient. We fully expect this result as both these agents generate their behavior by sampling from these distributions. It is also clear that the GT agent performs very differently from the human data, based on most of the

	Adaptive	DBA	DBRA	GT
d^O	0.57	0.008	0.008	33.26
d^R	0.21	0.0005	0.01	0.19
d^B	11.74	0.008	0.11	32.83
d^{Y_8}	4.22	16.34	20.10	97.02

Table 1. Similarity to human play, based on various metrics.

metrics. Naturally, the Adaptive Agent scores worse than the distribution-based agents on the temporally-independent distribution metrics d^O , d^R , and d^B , but its behavior is still relatively close to human behavior. On the temporally-dependent reciprocation-chain metric d^{Y_8} , the Adaptive Agent scores much better in similarity to the human traces.

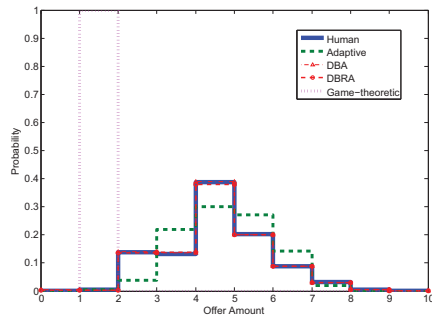
To get a more intuitive sense of the differences in the trace data, we also display the actual distributions that underlie the metrics in Figure 6 which shows the distributions of offer amounts for each of the agent types, the probability of rejection given each offer amount, the distribution of offer recipients, ordered from most likely to least likely, and the probability that an offer will be reciprocated, given that a chain of c offers have been made between the players in the past $c = 1, 2, \dots, 8$ time periods.

While the Adaptive Agent may not have been the most human-like agent according to the other three metrics, the form of its distributions still reasonably resembled the distributions produced by human play. However, on the time-dependent reciprocation-based metric, it is very clear that the Adaptive Agent is the only one that exhibits behavior that is similar to human play. This temporal dependence is crucial to creating agent behavior that emulates human behavior.

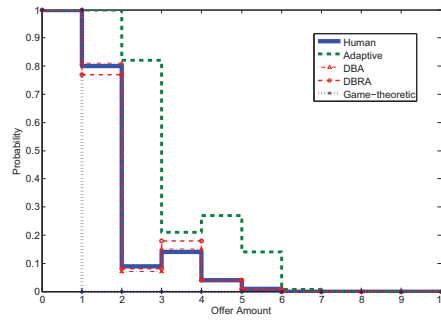
7 Related Work

Our choice to investigate the Ultimatum Game was motivated by its long history in the field and the fact that it is a leading example of where game-theoretic reasoning fails to predict consistent human behaviors [4, 10, 5]. Economists and sociologists have proposed many variants and contexts of the Ultimatum Game that seek to address the divergence between the “rational” Nash equilibrium strategy and observed human behavior, for example, examining the game when played in different cultures, with members of different communities, where individuals are replaced by groups, where the players are autistic, and when one of the players is a computer. Interestingly, isolated non-industrialized cultures, people who have studied economics, groups, being autistic, and playing with a computer all tend to lead to less cooperative behavior [4, 10, 8, 6, 1, 3]. Learning human game data is a promising approach for quickly learning realistic models of behavior. In the paper, we have demonstrated this approach in SUG, and proposed metrics that evaluate the similarity between autonomous agents’ game traces and human game traces.

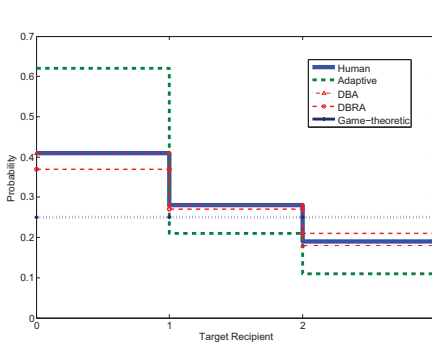
Recently, there has also been other work attempting to model human behavior in multi-agent scenarios, primarily in social network and other domains modeled by



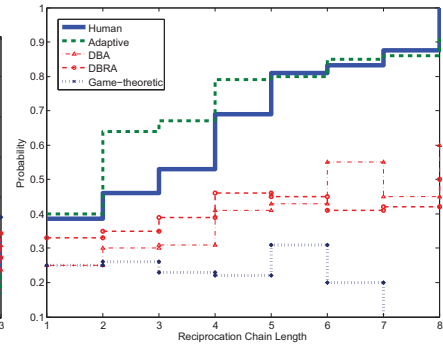
(a) Distribution of Offer Amounts



(b) Rejection Probabilities by Offer Amount



(c) Target Recipient Distribution



(d) Reciprocation Probabilities Given Reciprocation Chain of Length $c = 1, 2, \dots, 8$.

graphical relationship structures [7]. In contrast, our work focuses on multi-agent situations where motivated agents make sequential decisions, thus requiring models that include some consideration of utilities and their interplay with psychological effects. Our Adaptive Agent is a simple model, with parameters that are fit to the collected data, that demonstrates this approach.

Finally, a critical aspect of this line of work must include the development of appropriate metrics for evaluating the verisimilitude of the autonomous agent behaviors to human behavior. While there is a long literature on time-series metrics [9], in this paper, we show that these metrics do not capture the temporal causality patterns that are key to evaluating human behaviors, and thus are insufficient to evaluate agent behaviors when used alone.

8 Conclusion

Our goal is to develop approaches to create autonomous agents that replicate human behavior in multi-agent domains where humans make sequential decisions over time. To create and evaluate these agents, one needs appropriate metrics to characterize the deviations from the source behavior. The challenge is that a single source behavior in dynamic environments produces not a single decision but instead multiple traces where each trace is a sequence of decisions. A single source can produce a diverse collection of traces. Thus, the challenge is to find a way to compare collections of traces.

We introduced the Social Ultimatum Game and in that context, developed time-collapsed and time-dependent metrics to evaluate a collection of autonomous agents. We showed that agents built on time-collapsed metrics can miss key characteristics of human play, in particular an accurate model of temporal reciprocity. While our adaptive agent was able to perform closer to this metric, the key is the identification of time-dependent metrics as a key factor in evaluating emulation agents. This also has implications on the type of agent model necessary to have as a substrate upon which one can learn from human data.

Going forward, we will consider more complex domains and potential corresponding models. We will require both general, parameterized models that can be learned from data, as well as more formal methods for constructing appropriate temporal metrics to automatically evaluate the realism of the learned behaviors.

References

1. Carnevale, C.R.P.J.: Group choice in ultimatum bargaining. *Organizational Behavior and Human Decision Processes* 72(2), 256–279 (1997)
2. Chang, Y.H., Levinboim, T., Maheswaran, R.: The social ultimatum game. In: *Proceedings of the NIPS 2010 Workshop on Decision Making with Multiple Imperfect Decision Makers* (2010)
3. Frank, R.H., Gilovich, T., Regan, D.T.: Does studying economics inhibit cooperation? *The Journal of Economic Perspectives* 7(2), 159–171 (1993)
4. Henrich, J.: Does culture matter in economic behavior? ultimatum game bargaining among the machiguenga. *American Economic Review* 90(4), 973–979 (2000)

5. Henrich, J., Heine, S.J., Norenzayan, A.: The weirdest people in the world? *Behavioral and Brain Sciences* 33(2-3), 61–83 (2010)
6. Hill, E., Sally, D.: *Dilemmas and bargains: Theory of mind, cooperation and fairness* (2002), working paper, University College, London
7. Judd, S., Kearns, M., Vorobeychik, Y.: Behavioral dynamics and influence in networked coloring and consensus. In: *Proceedings of the National Academy of Science* (2010)
8. Mascha van't Wout, René S. Kahn, A.G.S., Aleman, A.: Affective state and decision-making in the ultimatum game. *Experimental Brain Research* 169(4), 564–568 (2006)
9. Mitsa, T.: *Temporal Data Mining*. CRC Press (2010)
10. Oosterbeek, H., Sloof, R., van de Kuilen, G.: Differences in ultimatum game experiments: Evidence from a meta-analysis. *Experimental Economics* 7, 171–188 (2004)

Non-Cooperative Bargaining with Arbitrary One-Sided Uncertainty

S. Ceppi and N. Gatti and C. Iuliano

Dipartimento di Elettronica e Informazione, Politecnico di Milano
Piazza Leonardo da Vinci 32, I-20133 Milano, Italy
{ceppi,ngatti,iuliano}@elet.polimi.it

Abstract. Non-cooperative bargaining is modeled as an extensive-form game with uncertain information and infinite actions. Its resolution is a long-standing open problem and no algorithm addressing uncertainty over multiple parameters is known. We provide an algorithm to solve bargaining with any kind of one-sided uncertainty. Our algorithm reduces a bargaining problem to a finite game, solves this last game, and then maps its strategies with the original continuous game. Computational complexity is polynomial with two types, while with more types the problem is hard and only small settings can be solved in exact way.

1 Introduction

The automation of economic transactions through negotiating software agents is receiving a large attention in the artificial intelligence community. Autonomous agents can lead to economic contracts more efficient than those drawn up by humans, saving also time and resources [13]. We focus on the main bilateral negotiation setting: the *bilateral bargaining*. This setting is characterized by the interaction of two agents, a *buyer* and a *seller*, who can cooperate to produce a utility surplus by reaching an economic agreement, but they are in conflict on what specific agreement to reach. The most expressive model for non-cooperative bargaining is the *alternating-offers* [10]: agents alternately act in turns and each agent can accept the offer made by her opponent at the previous turn or make a new offer. Agents' utility over the agreements depends on some parameters: *discount factor*, *deadline*, *reservation price*. In real-world settings, agents have a Bayesian prior over the values of the opponents' parameters.

The alternating-offers is an *infinite-horizon* (agents can indefinitely bargain) *extensive-form* (the game is sequential) *Bayesian* (information is uncertain) game and the number of available actions to each agent is infinite (an offer is a real value). The appropriate solution concept is the *sequential equilibrium* [7]. The game theoretic study of bargaining with uncertain information is an open challenging problem. No work presented in the literature so far is applicable regardless of the uncertainty *kind* (i.e., the uncertain parameters) and *degree* (i.e., the number of the parameters' possible values). Microeconomics provides analytical results for settings without deadlines, for single uncertainty kinds,

and with narrow degrees of uncertainty, e.g., over the discount factor of one agent with two possible values [11] and over the reservation price of both agents with two possible values per agent [1]. Computer science provides algorithms to search for sequential equilibria [8] only with finite games and without producing belief systems off the equilibrium path. This makes such algorithms not suitable for bargaining. Several efforts have been accomplished to extend the backward induction algorithm to solve games with uncertain information [3]. However, as shown in [4], the solutions produced by these algorithms may not be equilibria. Finally, the algorithm provided by [4] solves settings with one-sided uncertain deadlines, but its extension to general settings appears to be impractical due to the mathematical machinery it needs.

The work in [4] provides the unique known computational complexity result, showing that with one-sided uncertain deadlines the problem is polynomial in the length of the bargaining independently of the number of types. However, this uncertainty situation is very special because all the types have the same utility functions before their deadlines. This fact leads all the types whose deadline is not expired to have the same behavior, drastically reducing thus the complexity of the problem. When discount factors and reservation prices are uncertain, the types have different utility functions and we expect that they have different optimal behaviors. The difficulty of developing an exact algorithm for the bargaining problem pushed the scientific community to produce approximate solutions. A large number of tactic-based heuristics are available, e.g., see [2], but none provides bounds over the solution quality in terms of ϵ -Nash equilibrium.

In this paper, after having reviewed the alternating-offers protocol and its solution with complete information (Section 2), and after having discussed the model with uncertainty (Section 3), we present a sound and complete algorithm to solve settings with arbitrary kinds and degrees of uncertainty (Section 4). Our algorithm reduces the bargaining game to a finite game, solves this last game, and finally maps its equilibrium strategies to the original continuous game. We initially focus on settings with two possible types. We define a belief system $\bar{\mu}$ and a strategy profile $\bar{\sigma}$ where agents can make a finite number of offers and the randomization probabilities with which agents make such offers are parameters. To compute the values of these parameters such that $(\bar{\mu}, \bar{\sigma})$ is a sequential equilibrium, we build a finite game and we provide an algorithm based on support-enumeration to solve it. We show that the problem is polynomial in the deadline length. Then, we extend the algorithm to more than two types by exploiting mathematical programming and we experimentally evaluate it.

2 Bargaining Model and Complete Information Solution

We present the alternating-offers protocol [10] with deadlines. There are two agents, a buyer \mathbf{b} and a seller \mathbf{s} , who can play alternatively at discrete time points $t \in \mathbb{N}$. The function $\iota : \mathbb{N} \rightarrow \{\mathbf{b}, \mathbf{s}\}$ returns the agent that plays at time point t , and it is such that $\iota(t) \neq \iota(t+1)$. We study single-issue bargaining because our aim is the study of settings with uncertainty and algorithms for single-issue

problems can be easily extended to multi-issue problems as it is shown in [3]. Agents bargain on the value of a variable $x \in \mathbb{R}$, e.g., representing the price. The pure strategies $\sigma_{\iota(t)}(t)$ available to agent $\iota(t)$ at $t > 0$ are: *offer*(x), where x is the offer for the issue; *accept*, that concludes the bargaining with an agreement, formally denoted by (x, t) , where x is such that $\sigma_{\iota(t-1)}(t-1) = \text{offer}(x)$ (i.e., the value offered at $t-1$), and t is the time point at which the offer is accepted; *exit*, that concludes the bargaining with a disagreement, formally denoted by *NoAgreement*. At $t = 0$ only actions *offer*(x) and *exit* are available.

Seller's and buyer's utility functions, denoted by $U_{\mathbf{s}} : (\mathbb{R} \times \mathbb{N}) \cup \text{NoAgreement} \rightarrow \mathbb{R}$ and $U_{\mathbf{b}} : (\mathbb{R} \times \mathbb{N}) \cup \text{NoAgreement} \rightarrow \mathbb{R}$ respectively, return the agents' utility for each possible outcome. Each utility function depends on the following parameters: the reservation prices, denoted by $RP_{\mathbf{b}} \in \mathbb{R}^+$ and $RP_{\mathbf{s}} \in \mathbb{R}^+$ for buyer and seller respectively (we assume $RP_{\mathbf{b}} \geq RP_{\mathbf{s}}$), the discount factor, denoted by $\delta_{\mathbf{b}} \in (0, 1]$ and $\delta_{\mathbf{s}} \in (0, 1]$ for buyer and seller respectively, and the deadlines, denoted by $T_{\mathbf{b}} \in \mathbb{N}$ and $T_{\mathbf{s}} \in \mathbb{N}$ for buyer and seller respectively. The buyer's utility function is:

$$U_{\mathbf{b}}(\cdot) = \begin{cases} \text{NoAgreement} & 0 \\ (x, t) & \begin{cases} (RP_{\mathbf{b}} - x) \cdot (\delta_{\mathbf{b}})^t & \text{if } t \leq T_{\mathbf{b}} \\ \epsilon & \text{otherwise} \end{cases} \end{cases},$$

where $\epsilon < 0$ (after T_i , $U_i(x, t)$ is strictly negative and thus agent i strictly prefers to leave the game rather than reaching any agreement). The seller's utility function is analogous, except for $U_{\mathbf{s}}(x, t) = (x - RP_{\mathbf{s}}) \cdot (\delta_{\mathbf{s}})^t$ if $t \leq T_{\mathbf{s}}$.

With complete information, the appropriate solution concept is the *subgame perfect equilibrium*. The solution can be found by using backward induction as follows. We call $T = \min\{T_{\mathbf{b}}, T_{\mathbf{s}}\}$ and we call $x^*(t)$ the $\iota(t)$'s best offer at t , if this offer exists. It can be easily observed that the outcome of each subgame which starts at $t \geq T$ is *NoAgreement*, because at least one agent strictly prefers to exit the game rather than to reach any agreement. Now we consider the subgame which starts at $t = T - 1$. This subgame is essentially an *ultimatum game* [5]. $\iota(T)$ accepts any offer x such that $U_{\iota(T)}(x, T) \geq 0$ ($x \leq RP_{\mathbf{b}}$ if $\iota(T) = \mathbf{b}$ and $x \geq RP_{\mathbf{s}}$ if $\iota(T) = \mathbf{s}$), she leaves the game otherwise. The $\iota(T-1)$'s optimal offer $x^*(T-1)$ maximizes $\iota(T-1)$'s utility (i.e., $x^*(T-1) = RP_{\mathbf{b}}$ if $\iota(T-1) = \mathbf{s}$ and $x^*(T-1) = RP_{\mathbf{s}}$ if $\iota(T-1) = \mathbf{b}$). The subgames which start at time $t < T-1$ can be studied in a similar way. Suppose that we have found $x^*(t+1)$ and that we want to derive $x^*(t)$. We can consider the subgame composed of time points t and $t+1$ as an ultimatum game variation in which $\iota(t+1)$ accepts any offer x such that $U_{\iota(t+1)}(x, t+1) \geq U_{\iota(t+1)}(x^*(t+1), t+2)$ and offers $x^*(t+1)$ otherwise. The $\iota(t)$'s best offer, among all the acceptable offers at time point $t+1$, is the one which maximizes $\iota(t)$'s utility. We can compute this offer as:

$$x^*(t) = \begin{cases} RP_{\mathbf{s}} + (x^*(t+1) - RP_{\mathbf{s}}) \cdot \delta_{\mathbf{s}} & \text{if } \iota(t) = \mathbf{b} \\ RP_{\mathbf{b}} - (RP_{\mathbf{b}} - x^*(t+1)) \cdot \delta_{\mathbf{b}} & \text{if } \iota(t) = \mathbf{s} \end{cases}.$$

The computation of the values $x^*(t)$ is linear in t . We report the buyer's subgame perfect equilibrium strategies (the seller's ones are analogous):

$$\sigma_{\mathbf{b}}^*(t) = \begin{cases} t = 0 & \text{offer}(x^*(0)) \\ 0 < t < T & \begin{cases} \text{accept} & \text{if s's offer} \leq x^*(t-1) \\ \text{offer}(x^*(t)) & \text{otherwise} \end{cases} \\ t = T & \begin{cases} \text{accept} & \text{if s's offer} \leq x^*(t-1) \\ \text{exit} & \text{otherwise} \end{cases} \\ t > T & \text{exit} \end{cases}$$

3 Introducing Uncertainty

We consider one-sided uncertain settings where the buyer's parameters are uncertain to the seller (the reverse situation is analogous). Our game is an imperfect-information game in which the buyer can be of different types, each one with different values of $RP_{\mathbf{b}}$, $\delta_{\mathbf{b}}$, and $T_{\mathbf{b}}$. Uncertainty is over the actual type of the buyer. For the sake of presentation, we describe our algorithm for the basic case where the number of buyer's types is two, we call them \mathbf{b}_1 and \mathbf{b}_2 . Then, we discuss how to extend it with more than two types. Without loss of generality we assume $T_{\mathbf{b}_1} \leq T_{\mathbf{b}_2}$. We call $\mu(t) = (\Theta_{\mathbf{b}}(t), P_{\mathbf{b}}(t))$ the s's beliefs about \mathbf{b} 's type where $\Theta_{\mathbf{b}}(t) \in \wp(\{\mathbf{b}_1, \mathbf{b}_2\})/\emptyset$ and $P_{\mathbf{b}}(t) = \{\omega_{\mathbf{b}_1}(t), \omega_{\mathbf{b}_2}(t)\}$ (\wp denotes the power set and $\omega_{\mathbf{b}_i}(t)$ denotes the probability that \mathbf{b} 's type is \mathbf{b}_i at time t). $\mu(0)$ are data of the problem.

Example 31 *The parameters values are: $RP_s = 0$, $\delta_s = 0.75$, $T_s = 10$; $RP_{\mathbf{b}_1} = 1$, $\delta_{\mathbf{b}_1} = 0.7$, $T_{\mathbf{b}_1} = 5$; the \mathbf{b}_2 's parameters values are: $RP_{\mathbf{b}_2} = 0.9$, $\delta_{\mathbf{b}_2} = 0.8$, $T_{\mathbf{b}_2} = 5$. Assume that $\iota(0) = \mathbf{b}$ and that the values $\omega_{\mathbf{b}_1}(0)$ and $\omega_{\mathbf{b}_2}(0)$ are arbitrary.*

The appropriate solution concept is the sequential equilibrium [7]. It is a couple $a = (\mu, \sigma)$, also called assessment, in which μ is a belief system that specifies how agents must update their beliefs during the game and σ is the agents' strategy profile that specifies how they must act. μ must be *consistent* with σ and σ must be *sequentially rational* given μ . On the equilibrium path, μ is consistent to σ if it is equal to the beliefs derived from σ by using the Bayes rule. Off the equilibrium path, the Bayes rule is not applicable and two notions of consistency can be employed: *weak consistency* does not pose any constraint, while *strong consistency* requires that a sequence of fully mixed strategies exists such that its limit converges to σ and that the limit of the sequence of beliefs derived from the fully mixed strategies by using the Bayes rule converges to μ . In bargaining problems, strong consistency is commonly used because it allows one to exclude non reasonable equilibria. We remark that every game admits at least one strong sequential equilibrium. Off the equilibrium path we impose that is, if at time point t we have $\omega_{\mathbf{b}_i}(t) = 0$, then for any $\tau > t$ we keep $\omega_{\mathbf{b}_i}(\tau) = 0$.

4 The Algorithm

Since bargaining with uncertainty may not admit any equilibrium in pure strategies, as shown in [4], we directly search for equilibria in mixed strategies. The basic idea behind our work is to solve the bargaining problem by reducing it to a finite game, deriving equilibrium strategies such that on the equilibrium path

the agents can act only a finite set of actions, and then by searching for the agents' optimal strategies on the path. Our work is structured in the following three steps. 1) We analytically derive an assessment $\bar{a} = (\bar{\mu}, \bar{\sigma})$ in which the randomization probabilities of the agents are parameters and such that, when the parameters' values satisfy some conditions, \bar{a} is a sequential equilibrium. 2) We formulate the problem of finding the values of the agents' randomization probabilities in \bar{a} as the problem of finding a sequential equilibrium in a reduced bargaining game with finite actions, and we prove that there always exist values such that \bar{a} is a sequential equilibrium. 3) We develop an algorithm based on support enumeration to solve the reduced game when the types are two and we show that its computational complexity is polynomial in the agents' deadlines. Then we develop an algorithm based on linear complementarity mathematical programming to solve the case with more than two types.

4.1 Deriving Equilibrium Strategies

Without loss of generality, on the equilibrium path we study only time points $t < T_{\mathbf{b}_1}$. This is because, if agents reach time points $t \geq T_{\mathbf{b}_1}$ on the equilibrium path, then the bargaining at t is a game with complete-information in which agents are \mathbf{b}_2 and \mathbf{s} . Indeed, \mathbf{b}_1 never makes offers at time $t \geq T_{\mathbf{b}_1}$, action *exit* being the dominant action, and therefore, if action *offer*(x) is observed at $t \geq T_{\mathbf{b}_1}$, the Bayes rule imposes that $\omega_{\mathbf{b}_1}(t) = 0$. We build an assessment \bar{a} such that, on the equilibrium path, the $\iota(t)$'s offers at $t < T_{\mathbf{b}_1}$ belong to a finite set $X(t) := \{x_{\mathbf{b}_i}^*(t) : \forall i\}$, where $x_{\mathbf{b}_i}^*(t)$ is the $\iota(t)$'s optimal offer at t in the corresponding complete-information game between \mathbf{b}_i and \mathbf{s} computed as previously discussed. Offering at t any $x \notin X(t)$ does not allow $\iota(t)$ to improve her expected utility. In Fig. 1 we show $x_{\mathbf{b}_1}^*(t)$ and $x_{\mathbf{b}_2}^*(t)$ related to Example 31. We connect the offers $x_{\mathbf{b}_1}^*(t)$ with a dotted line and the offers $x_{\mathbf{b}_2}^*(t)$ with a dashed line.

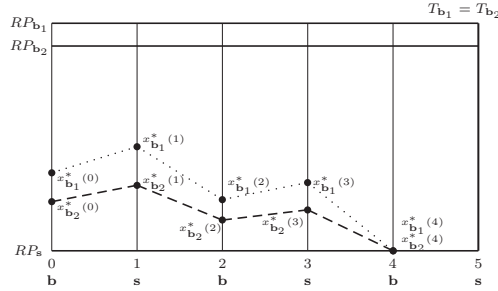


Fig. 1. $x_{\mathbf{b}_i}^*(t)$ in the complete information games between \mathbf{s} and \mathbf{b}_i (see Example 31).

We focus on \bar{a} . For each $t < T_{\mathbf{b}_1}$ we rank the values in $X(t)$ in increasing order and we call $\mathbf{b}_w = \arg \max_{i \in \{\mathbf{b}_1, \mathbf{b}_2\}} \{x_i^*(0)\}$ and $\mathbf{b}_s = \arg \min_{i \in \{\mathbf{b}_1, \mathbf{b}_2\}} \{x_i^*(0)\}$ where w means *weak* and s means *strong*. In Fig. 1 we have $\mathbf{b}_w = \mathbf{b}_1$ and $\mathbf{b}_s = \mathbf{b}_2$. According to [12], the adjectives ‘strong’ and ‘weak’ refer to the contractual power

of the corresponding buyer's type: in complete-information settings the seller's expected utility is larger when it bargains with \mathbf{b}_w rather than when it bargains with \mathbf{b}_s . In two cases, the type with the strongest contractual power at $t = 0$ is not the strongest for all $t > 0$. This happens, first, when there exists at least a time point t where $x_{\mathbf{b}_s}^*(t) > x_{\mathbf{b}_w}^*(t)$, second, when $T_{\mathbf{b}_w} > T_{\mathbf{b}_s}$. These two cases represent two exceptions that can be easily tackled by modifying the computation of $x_{\mathbf{b}_s}^*(t)$ and $x_{\mathbf{b}_w}^*(t)$. For reasons of space, we omit their description. The basic idea behind \bar{a} is that, when agents are forced to make the offers in $X(t)$, \mathbf{b}_w can gain utility from disguising herself as \mathbf{b}_s , making the optimal \mathbf{b}_s 's offers, while \mathbf{b}_s prefers to signal her own type, making offers different from the \mathbf{b}_w 's ones. That is, \mathbf{b}_w acts in order to increase her expected utility with respect to the situation where \mathbf{s} believes \mathbf{b} 's type to be \mathbf{b}_w with certainty. The same idea is used in [1].

We focus on the buyer's behaviour. On the equilibrium path, \mathbf{b}_w randomizes between offering $x_{\mathbf{b}_w}^*(t)$ (or, equivalently, accepting $x_{\mathbf{b}_w}^*(t-1)$) and offering $x_{\mathbf{b}_s}^*(t)$ (the offer $x_{\mathbf{b}_s}^*(t-1)$ is always accepted, leading to the largest possible utility), whereas \mathbf{b}_s offers $x_{\mathbf{b}_s}^*(t)$ in pure strategies (or, equivalently, accepts $x_{\mathbf{b}_s}^*(t-1)$). We denote by $1 - \alpha(t)$ and $\alpha(t)$ the \mathbf{b}_w 's randomization probabilities over offering $x_{\mathbf{b}_w}^*(t)$ /accepting $x_{\mathbf{b}_w}^*(t-1)$ and offering $x_{\mathbf{b}_s}^*(t)$, respectively, and we consider $\alpha(t)$ as parameters. We remark that, if $\alpha(t) = 1$, then the strategies of \mathbf{b}_w and \mathbf{b}_s are pure and they are the same. On the equilibrium path, the beliefs are updated according to the Bayes rule. We call $\omega_{\mathbf{b}_i}^*(t)$ the probability over type \mathbf{b}_i at time t produced according to the Bayes rule after that \mathbf{b} made *offer*($x_{\mathbf{b}_s}^*(t-1)$) at time $t-1$. We have $\omega_{\mathbf{b}_s}^*(t) = \frac{\omega_{\mathbf{b}_s}^*(t-1)}{\alpha(t-1)\omega_{\mathbf{b}_w}^*(t-1) + \omega_{\mathbf{b}_s}^*(t-1)}$ and $\omega_{\mathbf{b}_w}^*(t) = 1 - \omega_{\mathbf{b}_s}^*(t)$. We notice that when the strategies are pure, if $\alpha(t-1) = 1$, then $\omega_{\mathbf{b}_w}^*(t) = \omega_{\mathbf{b}_w}^*(t-1)$ and $\omega_{\mathbf{b}_s}^*(t) = \omega_{\mathbf{b}_s}^*(t-1)$, while, if $\alpha(t-1) = 0$, then $\omega_{\mathbf{b}_w}^*(t) = 0$ and $\omega_{\mathbf{b}_s}^*(t) = 1$.

To characterize \mathbf{b} 's strategies off the equilibrium path, at each time t we divide the domain of x as: $\mathcal{D}1 := (x_{\mathbf{b}_w}^*(t-1), +\infty)$, $\mathcal{D}2 := (x_{\mathbf{b}_s}^*(t-1), x_{\mathbf{b}_w}^*(t-1)]$, $\mathcal{D}3 := (-\infty, x_{\mathbf{b}_s}^*(t-1)]$. We call y the value such that $\sigma_{\mathbf{s}}(t-1) = \text{offer}(y)$. The \mathbf{b}_w 's strategies are: if $y \in \mathcal{D}1$, then y is rejected; if $y \in \mathcal{D}2$, then y is accepted with probability of $1 - \alpha(t)$ and rejected to offer $x_{\mathbf{b}_s}^*$ otherwise, and, if $y \in \mathcal{D}3$, then the offer is accepted (no better agreement can be reached from time point $t+1$ on). The \mathbf{b}_s 's strategies are exactly her optimal strategies in the complete-information game between \mathbf{b}_s and \mathbf{s} : if $y \in \mathcal{D}1$ or $y \in \mathcal{D}2$, then the offer is refused and, if $y \in \mathcal{D}3$, then the offer is accepted. We notice that, if $\alpha(t) = 1$, then \mathbf{b}_w and \mathbf{b}_s have the same strategies also off the equilibrium path. Formally, the strategies are (at $t > T_{\mathbf{b}_1}$ the buyer's strategies are those with complete information; the strategies in the case in which the buyer's type is \mathbf{b}_s and $\omega_{\mathbf{b}_s}(t) = 0$ are):

$$\sigma_{\mathbf{b}_w}^*(t) = \begin{cases} t = 0 & \begin{cases} \text{offer}(x_{\mathbf{b}_w}^*(0)) & 1 - \alpha(0) \\ \text{offer}(x_{\mathbf{b}_s}^*(0)) & \alpha(0) \end{cases} \\ 0 < t < T_{\mathbf{b}_1} & \begin{cases} y \in \mathcal{D}1 & \begin{cases} \text{offer}(x_{\mathbf{b}_w}^*(t)) & 1 - \alpha(t) \\ \text{offer}(x_{\mathbf{b}_s}^*(t)) & \alpha(t) \end{cases} \\ y \in \mathcal{D}2 & \begin{cases} \text{accept} & 1 - \alpha(t) \\ \text{offer}(x_{\mathbf{b}_s}^*(t)) & \alpha(t) \end{cases} \\ y \in \mathcal{D}3 & \text{accept} \end{cases} \end{cases}, \sigma_{\mathbf{b}_s}^*(t) = \begin{cases} t = 0 & \text{offer}(x_{\mathbf{b}_s}^*(0)) \\ 0 < t < T_{\mathbf{b}_1} & \begin{cases} y \in \mathcal{D}1, \mathcal{D}2 & \text{offer}(x_{\mathbf{b}_s}^*(t)) \\ y \in \mathcal{D}3 & \text{accept} \end{cases} \end{cases}.$$

To characterize the beliefs and \mathbf{s} 's strategies off the equilibrium path, at each time t we divide the domain of x as: $\mathcal{D}1' := [x_{\mathbf{b}_w}^*(t-1), +\infty)$, $\mathcal{D}2' := [x_{\mathbf{b}_s}^*(t-1), x_{\mathbf{b}_w}^*(t-1))$, $\mathcal{D}3' := (-\infty, x_{\mathbf{b}_s}^*(t-1))$. We call y the value such that $\sigma_{\mathbf{b}}(t-1) = \text{offer}(y)$. If $\omega_{\mathbf{b}_w}(t-1) > 0$, then the beliefs are: if $y \in \mathcal{D}1'$, then \mathbf{b} is believed \mathbf{b}_w with a probability of 1; if $y \in \mathcal{D}3'$, then the probabilities of \mathbf{b} 's types are the same that we compute on the equilibrium path when $y = x_{\mathbf{b}_s}^*(t-1)$; if $y \in \mathcal{D}2'$, then the \mathbf{b}_s 's probability increases linearly in y such that, if y goes to $x_{\mathbf{b}_w}^*(t-1)$, then $\omega_{\mathbf{b}_s}(t)$ goes to 0 and, if y goes to $x_{\mathbf{b}_s}^*(t-1)$, then $\omega_{\mathbf{b}_s}(t)$ goes to $\omega_{\mathbf{b}_s}^*(t)$ (notice that we cannot use '=' , since the cases with '=' are on the equilibrium path). Defining $\kappa(t, y) = \frac{x_{\mathbf{b}_w}^*(t) - y}{x_{\mathbf{b}_w}^*(t) - x_{\mathbf{b}_s}^*(t)}$, the belief system is:

$$\bar{\mu}(t) = \begin{cases} y \in \mathcal{D}1' & \omega_{\mathbf{b}_s}(t) = 0 \\ y \in \mathcal{D}2' & \omega_{\mathbf{b}_s}(t) = \omega_{\mathbf{b}_s}^*(t) \kappa(t-1, y) \\ y \in \mathcal{D}3' & \omega_{\mathbf{b}_s}(t) = \omega_{\mathbf{b}_s}^*(t) \end{cases}.$$

We focus on the seller's behaviour. On the equilibrium path, \mathbf{s} randomizes between offering $x_{\mathbf{b}_s}^*(t)$ (or, equivalently, accepting $x_{\mathbf{b}_s}^*(t-1)$) and offering $x_{\mathbf{b}_w}^*(t)$ (the offer $x_{\mathbf{b}_w}^*(t-1)$ is always accepted, leading to the largest possible utility). We denote by $\beta(t)$ and $1 - \beta(t)$ the \mathbf{s} 's randomization probabilities over offering $x_{\mathbf{b}_s}^*(t)$ /accepting $x_{\mathbf{b}_s}^*(t-1)$ and offering $x_{\mathbf{b}_w}^*(t)$, respectively, and we consider $\beta(t)$ as parameters. Off the equilibrium path, the \mathbf{s} 's strategies are: if $y \in \mathcal{D}1'$, then the offer is accepted; if $y \in \mathcal{D}2'$, then the acceptance probability decreases linearly in y such that, if y goes to $x_{\mathbf{b}_w}^*(t-1)$, then it goes to 1 and, if y goes to $x_{\mathbf{b}_s}^*(t-1)$, then it goes to $\beta(t)$ (the \mathbf{s} 's probability to offer $x_{\mathbf{b}_w}^*(t)$ is 1 minus the acceptance probability); if $y \in \mathcal{D}3'$, then it is rejected to offer $x_{\mathbf{b}_w}^*(t)$ if $\beta(t) < 1$ and $x_{\mathbf{b}_s}^*(t)$ otherwise. Formally the strategies are (at $t > T_{\mathbf{b}_1}$ the seller's strategies are those with complete information):

$$\sigma_{\mathbf{s}}^*(t) = \begin{cases} t = 0 & \begin{cases} \text{offer}(x_{\mathbf{b}_w}^*(0)) & 1 - \beta(0) \\ \text{offer}(x_{\mathbf{b}_s}^*(0)) & \beta(0) \end{cases} \\ 0 < t < T_{\mathbf{b}_1} & \begin{cases} y \in \mathcal{D}1' & \text{accept} \\ y \in \mathcal{D}2' & \begin{cases} \text{offer}(x_{\mathbf{b}_w}^*(t)) & \kappa(t-1, y)(1 - \beta(t)) \\ \text{accept} & 1 - \kappa(t-1, y)(1 - \beta(t)) \end{cases} \\ y \in \mathcal{D}3' & \begin{cases} \text{offer}(x_{\mathbf{b}_w}^*(t)) & [1 - \beta(t)] \\ \text{offer}(x_{\mathbf{b}_s}^*(t)) & [\beta(t)] \end{cases} \end{cases} \end{cases}.$$

Call $\bar{\sigma} = (\sigma_{\mathbf{b}_w}^*, \sigma_{\mathbf{b}_s}^*, \sigma_{\mathbf{s}}^*)$. We state the following theorem.

Theorem 41 *If $\alpha(t), \beta(t) \in [0, 1]$ are such that, limited to the offers in $X(t)$, $\bar{\sigma}$ is sequentially rational given $\bar{\mu}$, then $\bar{a} = (\bar{\mu}, \bar{\sigma})$ is a sequential equilibrium.*

Proof. We assume that there are values $\alpha(t), \beta(t) \in [0, 1]$ such that, limited to the offers in $X(t)$, $\bar{\sigma}$ is sequentially rational given $\bar{\mu}$ and we prove: (i) sequential rationality off the equilibrium path and (ii) Kreps and Wilson's consistency. (The computation of the values of $\alpha(t), \beta(t)$ is discussed in the following sections.)

To prove (i) we need to show that agents cannot gain more by making offers not belonging to $X(t)$. At first, we characterize agents' strategies on the equilibrium path because it is useful for our proof. We do not consider the trivial cases in which $\omega_{\mathbf{b}_w}(0) = 1$ or $\omega_{\mathbf{b}_s}(0) = 1$; they can be solved as complete-information

games. It can be easily observed that if $\omega_{\mathbf{b}_w}(0) < 1$ then $\alpha(t) > 0$ for every t . Indeed, let suppose $\iota(0) = \mathbf{b}$ and $\omega_{\mathbf{b}_w}(0) < 1$, if $\alpha(0) = 0$, then \mathbf{b}_w and \mathbf{b}_s make different offers at time $t = 0$ and \mathbf{s} accepts both of them at $t = 1$. In this case \mathbf{b}_w can increase her utility acting as \mathbf{b}_s . Thus, two situations are possible: either $0 < \alpha(t) < 1$ or $\alpha(t) = 1$. If $0 < \alpha(t) < 1$, then \mathbf{b}_w randomizes between offering $x_{\mathbf{b}_w}(t)$ and $x_{\mathbf{b}_s}(t)$, so necessarily $0 < \beta(t+1) < 1$ because the game is non-degenerate. Otherwise, if $\alpha(t) = 1$, then necessarily $\beta(t+1) = 1$ because the game is non-degenerate and the case $\beta(t+1) = 0$ cannot lead to an equilibrium (\mathbf{b}_w can increase her utility by offering $x_{\mathbf{b}_w}^*(t)$ that will be always accepted).

Now, we are in the position to prove sequential rationality off the equilibrium path. We focus on the case $0 < \alpha(t) < 1$ and $0 < \beta(t+1) < 1$. We consider \mathbf{b}_w . Offering any $x > x_{\mathbf{b}_w}^*(t)$ at time t is dominated by offering $x_{\mathbf{b}_w}^*(t)$ because all these offers are accepted with a probability of one and $x_{\mathbf{b}_w}^*(t)$ gives a larger utility to \mathbf{b}_w . By construction, all the offers $x_{\mathbf{b}_s}^*(t) < x < x_{\mathbf{b}_w}^*(t)$ give to \mathbf{b}_w the same expected utility and all the offers $x < x_{\mathbf{b}_s}^*(t)$ are rejected, so the \mathbf{b}_w 's expected utility cannot be increased by performing them. In a similar way, it is possible to analyze the strategies of \mathbf{b}_s and \mathbf{s} . In the case of \mathbf{s} , if she acts at $t = 0$ or $t > 0$ after that \mathbf{b} makes an off-equilibrium-path offer, her strategy will be pure. It can be shown that, if $\sigma_{\mathbf{b}}(t-1) = \text{offer}(x)$ with $x < x_{\mathbf{b}_s}^*(t-1)$ and $\beta(t) < 1$, then \mathbf{s} 's optimal action is to offer $x_{\mathbf{b}_w}^*(t)$. Therefore, agents cannot gain more by making offers not belonging to $X(t)$.

In order to prove (ii), we need to provide a fully mixed strategy $\sigma_{\mathbf{b}_i,n}(t)$ such that $\lim_{n \rightarrow \infty} \sigma_{\mathbf{b}_i,n}(t) = \sigma_{\mathbf{b}_i}^*(t)$ and $\lim_{n \rightarrow \infty} \omega_{\mathbf{b}_i,n}(t) = \omega_{\mathbf{b}_i}(t)$ where $\omega_{\mathbf{b}_i,n}(t)$ are the sequences of beliefs derived from $\sigma_{\mathbf{b}_i,n}(t)$ by Bayes rule and $\omega_{\mathbf{b}_i}(t)$ are the beliefs prescribed by $\mu(t)$. The fully mixed strategies are:

$$\sigma_{\mathbf{b}_w,n}(t) = \begin{cases} y > x_{\mathbf{b}_w}^*(t) & \frac{1}{n} \\ y = x_{\mathbf{b}_w}^*(t) & 1 - \alpha(t) - A(n) \\ x_{\mathbf{b}_w}^*(t) > y > \bar{y} & \frac{1}{n} \\ \bar{y} \geq y > x_{\mathbf{b}_s}^*(t) & 1 - (1 - \alpha(t)) \frac{y - x_{\mathbf{b}_s}^*(t)}{n(\bar{y} - x_{\mathbf{b}_s}^*(t))} \cdot \sigma_{\mathbf{b}_s,n}(t) \\ y = x_{\mathbf{b}_s}^*(t) & \alpha(t) - A(n) \\ x_{\mathbf{b}_s}^*(t) > y & \frac{\alpha(t)}{n} \end{cases} = \begin{cases} y \geq x_{\mathbf{b}_w}^*(t) & \frac{1}{n} \\ x_{\mathbf{b}_w}^*(t) > y > \bar{y} & \frac{x_{\mathbf{b}_w}^*(t) - y}{n(x_{\mathbf{b}_w}^*(t) - \bar{y})} \\ \bar{y} \geq y > x_{\mathbf{b}_s}^*(t) & \frac{1}{n} \\ y = x_{\mathbf{b}_s}^*(t) & 1 - B(n) \\ x_{\mathbf{b}_s}^*(t) > y & \frac{1}{n} \end{cases},$$

where $A(n)$ and $B(n)$ are functions of n such that they go to zero as n goes to infinity and the sum over the probabilities of all actions is one. \square

4.2 Building the Reduced Bargaining Game

The previous section drastically reduces the complexity of solving a bargaining game, leaving open only the determination of the values of the randomization probabilities such that Theorem 41 holds. In this section, we formulate the problem of computing these values as the problem of solving a reduced bargaining game with finite actions. Since each finite game admits at least one equilibrium strategy, there always exist values such that Theorem 41 holds.

To compute the values of $\alpha(t)$ and $\beta(t)$ we “extract” the equilibrium path prescribed by assessment $\bar{\tau}$ given in the previous section. We build an imperfect-information extensive-form game with finite actions. It can be represented by

a game tree built as follows. Fig. 2 depicts the tree related to Example 31; for the sake of simplicity, we denote *accept* by ‘ A ’ and *offer*(x) by ‘ x ’; A' and A'' label two different A s of the same buyer’s type at the same t . In the root of the tree, *nature* plays drawing the buyer’s type: \mathbf{b}_1 or \mathbf{b}_2 with probability $\omega_{\mathbf{b}_1}(0)$ and $\omega_{\mathbf{b}_2}(0)$, respectively. Since the game is with imperfect information, \mathbf{s} cannot distinguish whether her opponent’s type is \mathbf{b}_1 or \mathbf{b}_2 unless she observes an action that can be made only by \mathbf{b}_1 or by \mathbf{b}_2 , respectively (e.g., in Fig. 2, action $x_{\mathbf{b}_1}^*(0)$ can be accomplished only by \mathbf{b}_1). Customarily in game theory, decision nodes that an agent cannot distinguish constitute an *information set* (in Fig. 2, dashed lines connect decision nodes of the same information set).

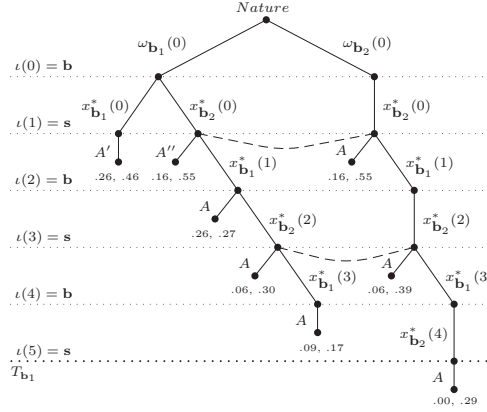


Fig. 2. Tree of the reduced game related to Example 31. We denote *accept* by A and *offer*(x) by value x . We report utilities $U_s(x, t), U_b(x, t)$ under the terminal nodes.

Let be $t = 0$. If $\iota(0) = \mathbf{b}$, the available actions are *offer*(x) with $x \in X(0)$ if $\mathbf{b}_i = \mathbf{b}_w$ and $x = x_{\mathbf{b}_s}^*(0)$ if $\mathbf{b}_i = \mathbf{b}_s$ (we recall that in Example 31, $\mathbf{b}_1 = \mathbf{b}_w$ and $\mathbf{b}_2 = \mathbf{b}_s$). When $\iota(0) = \mathbf{s}$, the available actions are *offer*(x) with $x \in X(0)$.

Let be $0 < t < T_{\mathbf{b}_1} - 1$. Suppose $\iota(t) = \mathbf{b}$. If $\mathbf{b}_i = \mathbf{b}_w$ and $\sigma_s(t-1) = \text{offer}(x_{\mathbf{b}_s}^*(t-1))$, then the only possible action is *accept*, otherwise, if $\sigma_s(t-1) = \text{offer}(x_{\mathbf{b}_w}^*(t-1))$, the available actions are *accept* and *offer*($x_{\mathbf{b}_s}^*(t)$). Action *accept* at time t leads to a terminal node in which the agents reach the agreement (x, t) where x is such that $\sigma_{\iota(t-1)}(t-1) = \text{offer}(x)$. In Fig. 2, under the terminal nodes, we report the agents’ utilities $U_s(x, t), U_b(x, t)$. If $\mathbf{b}_i = \mathbf{b}_s$ and $\sigma_s(t-1) = \text{offer}(x_{\mathbf{b}_s}^*(t-1))$ then the only possible action is *accept*, otherwise, if $\sigma_s(t-1) = \text{offer}(x_{\mathbf{b}_w}^*(t-1))$, the only available action is *offer*($x_{\mathbf{b}_s}^*(t)$). Suppose $\iota(t) = \mathbf{s}$. If $\sigma_b(t-1) = \text{offer}(x_{\mathbf{b}_w}^*(t-1))$ then the only possible action is *accept*, otherwise, if $\sigma_b(t-1) = \text{offer}(x_{\mathbf{b}_s}^*(t-1))$, the available actions are *accept* and *offer*($x_{\mathbf{b}_w}^*(t)$).

Let be $t = T_{\mathbf{b}_1} - 1$. If $x_{\mathbf{b}_w}^*(t) > x_{\mathbf{b}_s}^*(t)$, the tree building rules are those described for the previous case. Otherwise, if $x_{\mathbf{b}_w}^*(t) = x_{\mathbf{b}_s}^*(t)$, when $\mathbf{b}_i = \mathbf{b}_w$ and $\sigma_s(t-1) = \text{offer}(x_{\mathbf{b}_w}^*(t))$ the only available action is *accept*, as in Example 31 (see

Fig. 1). There cannot be any equilibrium when a buyer's type \mathbf{b}_i randomizes at t over accepting and offering offers of the same sequence of offers $x_{\mathbf{b}_i}^*(t)$.

We notice that the size of the tree is linear in $T_{\mathbf{b}_1}$. The values of $\alpha(t)$ and $\beta(t)$ can be computed finding a sequential equilibrium in the above reduced bargaining problem. By definition, the value of $\alpha(t)$ is equal to the probability with which \mathbf{b}_w makes *offer*($x_{\mathbf{b}_s}^*(t)$) at t in the reduced bargaining game, while the value of $1 - \beta(t)$ is equal to the probability with which \mathbf{s} makes *offer*($x_{\mathbf{b}_w}^*(t)$) at t in the reduced bargaining game. Since any finite game admits at least one sequential equilibrium, there always exist values of $\alpha(t)$ and $\beta(t)$ such that \bar{a} is a strong sequential equilibrium, namely, Theorem 41 always holds.

4.3 Solving the Reduced Bargaining Game

To compute an equilibrium, at first we represent the game in the sequence form [6] where agents' actions are sequences in the game tree.

The sequence form is represented with a sparse matrix in which the agent i 's actions are the sequences of her extensive form actions connecting the root of the tree to any information set of i . To avoid confusion, we shall use 'sequence' for the actions of the sequence form and 'action' for the actions of the extensive-form. For the sake of simplicity, we define different \mathbf{b} 's sequences for each different type. Considering the game tree reported in Fig. 2, the set of agent i 's sequences Q_i is: $Q_s = \{\emptyset, \langle A' \rangle, \langle A'' \rangle, \langle x_{\mathbf{b}_1}^*(1) \rangle, \langle x_{\mathbf{b}_1}^*(1), A \rangle, \langle x_{\mathbf{b}_1}^*(1), x_{\mathbf{b}_1}^*(3) \rangle, \langle x_{\mathbf{b}_1}^*(1), x_{\mathbf{b}_1}^*(3), A \rangle\}$, $Q_{\mathbf{b}_1} = \{\emptyset, \langle x_{\mathbf{b}_1}^*(0) \rangle, \langle x_{\mathbf{b}_2}^*(0) \rangle, \langle x_{\mathbf{b}_2}^*(0), A \rangle, \langle x_{\mathbf{b}_2}^*(0), x_{\mathbf{b}_2}^*(2) \rangle, \langle x_{\mathbf{b}_2}^*(0), x_{\mathbf{b}_2}^*(2), A \rangle\}$, $Q_{\mathbf{b}_2} = \{\emptyset, \langle x_{\mathbf{b}_2}^*(0) \rangle, \langle x_{\mathbf{b}_2}^*(0), x_{\mathbf{b}_2}^*(2) \rangle, \langle x_{\mathbf{b}_2}^*(0), x_{\mathbf{b}_2}^*(2), x_{\mathbf{b}_2}^*(4) \rangle\}$; where \emptyset is the empty sequence. Given two sequences q and q' with $q \in Q_{\mathbf{b}_i}$ and $q' \in Q_s$, the payoffs are non-null only if the node reached performing the combination of sequences q and q' is a terminal node. Let consider the subtree of type \mathbf{b}_1 shown in Fig. 2. The node reached performing $q = \langle x_{\mathbf{b}_2}^*(0) \rangle$ and $q' = \langle x_{\mathbf{b}_1}^*(1) \rangle$ is a non terminal node and, therefore, the payoffs are null, whereas the node reached performing $q = \langle x_{\mathbf{b}_2}^*(0) \rangle$ and $q' = \langle A'' \rangle$ is a terminal node and the payoffs are $U_s = 0.16$ and $U_b = 0.55$. We show in Table 1 the payoff bimatrix for \mathbf{b}_1 and \mathbf{s} (for reason of space we omit the empty sequences \emptyset). The payoff bimatrix for \mathbf{b}_2 and \mathbf{s} is defined similarly.

\mathbf{b}_1, \mathbf{s}	A'	A''	$x_{\mathbf{b}_1}^*(1)$	$x_{\mathbf{b}_1}^*(1), A$	$x_{\mathbf{b}_1}^*(1), x_{\mathbf{b}_1}^*(3)$
$x_{\mathbf{b}_1}^*(0)$.26, .46	-	-	-	-
$x_{\mathbf{b}_2}^*(0)$	-	.16, .55	-	-	-
$x_{\mathbf{b}_2}^*(0), A$	-	-	.26, .27	-	-
$x_{\mathbf{b}_2}^*(0), x_{\mathbf{b}_2}^*(2)$	-	-	-	.06, .30	-
$x_{\mathbf{b}_2}^*(0), x_{\mathbf{b}_2}^*(2), A$	-	-	-	-	.09, .17

Table 1. Payoff bimatrix for \mathbf{b}_1 and \mathbf{s} .

The sequence form presents some constraints over the probabilities with which the sequences are played by agents. We denote by $p_i(q)$ the probabil-

ity with which agent i makes sequence q . The constraints on the probabilities of the empty sequences are (by convention, we set $\omega_{\mathbf{s}}(0) = 1$):

$$p_i(\emptyset) = \omega_i(0) \quad \forall i, \quad (1)$$

constraints on the probabilities of non-empty sequences are:

$$p_i(q) = \sum_{a \text{ at } h_q} p_i(q|a) \quad \forall i, q \in Q_i, h_q \in I_q, \quad (2)$$

where I_q is the set of information sets reachable performing q , h_q is a information set belonging to I_q , a is an action available at information set h_q , and $q|a$ is the sequence obtained by adding action a to sequence q . Let consider \mathbf{s} in Fig. 2, if $q = \langle x_{\mathbf{b}_1}^*(1) \rangle$, then the constraint (2) is $p_{\mathbf{s}}(q) = p_{\mathbf{s}}(\langle x_{\mathbf{b}_1}^*(1), A \rangle) + p_{\mathbf{s}}(\langle x_{\mathbf{b}_1}^*(1), x_{\mathbf{b}_1}^*(3) \rangle)$, because only one information set is reachable by performing q . The values of $\alpha(t)$ and $\beta(t)$ are easily computable on the basis of probability $p_i(q)$. More precisely, called q a \mathbf{b}_w 's sequence that ends at time point $t - 1$ with $\iota(t) = \mathbf{b}$, we have $\alpha(t) = \frac{p_{\mathbf{b}_w}(q|x_{\mathbf{b}_s}^*(t))}{p_{\mathbf{b}_w}(q)}$. The values of $\beta(t)$ can be computed on the basis of $p_{\mathbf{s}}(q)$ in a similar way.

To solve the game we use the PNS algorithm [9] because it results very efficient: we can safely check a very small number of supports.

Theorem 42 *Excluded the degenerate case $\omega_{\mathbf{b}_w}(0) = 1$, agents' Nash equilibrium strategies on the equilibrium path in the reduced bargaining game are: if $\iota(0) = \mathbf{b}$, either \mathbf{b}_w 's and \mathbf{s} 's strategies are fully mixed or \mathbf{b}_w makes offer($x_{\mathbf{b}_s}^*(t)$) with probability of 1 at $t = 0$ and \mathbf{s} makes accept with probability of 1 at $t = 1$; if $\iota(0) = \mathbf{s}$, either \mathbf{s} makes offer($x_{\mathbf{b}_w}^*(0)$) with probability of 1 at $t = 0$ and from $t = 1$ on \mathbf{b}_w 's and \mathbf{s} 's strategies are fully mixed or \mathbf{s} makes offer($x_{\mathbf{b}_s}^*(0)$) with probability of 1 at $t = 0$ and \mathbf{b}_w makes accept at $t = 1$.*

Proof. We show that on the equilibrium path \mathbf{b}_w cannot make *accept* at time t with probability of 1 in all the decision nodes where multiple actions are available. Assume by contradiction that \mathbf{b}_w makes it. Then, \mathbf{s} 's best response is to make *accept* at time $t + 1$ with probability of 1. However, if \mathbf{s} makes such action at $t + 1$, \mathbf{b}_w 's best response is to make offer($x_{\mathbf{b}_s}^*(t)$) at time t and thus we have a contradiction. We show that on the equilibrium path \mathbf{b}_w cannot make offer($x_{\mathbf{b}_s}^*(t)$) at time $t > 0$ with probability of 1 in all the decision nodes where multiple actions are available. Assume by contradiction that it happens. Then, \mathbf{s} 's best response is to make *accept* at time $t - 1$ with probability of 1. Therefore, time point t would never be reached on the equilibrium path and then we have a contradiction.

The same above reasoning can be applied to show that on the equilibrium path \mathbf{s} cannot make with probability of 1 neither *accept* at time $t > 1$ nor offer($x_{\mathbf{b}_w}^*(t)$) at time $t > 0$. Thus, the unique possible agents' strategies on the equilibrium path are those reported in the theorem. If the fully mixed strategy is a Nash equilibrium, then it is by definition a sequential equilibrium. This is because every action is played with positive probability. If it is not an equilibrium, then necessarily the game concludes at $t = 1$. \square

The above theorem shows that for each bargaining problem we need to check only one joint support: if the fully mixed strategy is not an equilibrium, then on the equilibrium path the game concludes at $t = 1$. In this second case, to compute agents' equilibrium strategies off the equilibrium path it is sufficient to solve the reduced bargaining game from $t \geq 2$ with initial beliefs. The computational complexity of finding agents' equilibrium strategies on the equilibrium path is polynomial in $T_{\mathbf{b}_1}$, because the computational complexity of solving a linear feasibility problem is polynomial in the number of variables, this last number rises linearly in $T_{\mathbf{b}_1}$, and the number of joint supports to be checked is constant in the size of the game. Off the equilibrium path a number of joint supports that rises linearly in $T_{\mathbf{b}_1}$ must be checked, then the computational complexity is polynomial in $T_{\mathbf{b}_1}$. We use AMPL and CPLEX to solve the game. The computational times are negligible (< 1 s) even for large problems (up to $T_{\mathbf{b}_1} = 500$) with a 2.33 GHz 8 GB RAM UNIX computer. We report in Tab. 2 the values of $\alpha(t)$ and $\beta(t)$ for Example 31 with different values of initial beliefs.

$\omega_{\mathbf{b}_1}(0)$	$\omega_{\mathbf{b}_2}(0)$	$\alpha(0)$	$\beta(1)$	$\alpha(2)$	$\beta(3)$
0.10	0.90	1.00	1.00	1.00	1.00
0.70	0.30	1.00	1.00	0.86	0.77
0.80	0.20	0.68	0.69	0.44	0.77

Table 2. Values of $\alpha(t)$ s and $\beta(t)$ s. When $\omega_{\mathbf{b}_1}(0) = 0.1$ and $\omega_{\mathbf{b}_1}(0) = 0.7$ players always act in pure strategies; when $\omega_{\mathbf{b}_1}(0) = 0.8$ players randomize.

4.4 Extension to More than Two Types

Here the idea is the same of the two-type solution. At first, we compute all the sequences of optimal offers $x_{\mathbf{b}_i}^*(t)$ in the complete-information games between \mathbf{b}_i and \mathbf{s} . We rank the buyer's types from the strongest to the weakest according to $x_{\mathbf{b}_i}^*(0)$. At t each buyer's type \mathbf{b}_i randomizes over all the offers $x_{\mathbf{b}_j}^*(t)$ such that \mathbf{b}_j is not weaker than \mathbf{b}_i and \mathbf{b}_j is believed by \mathbf{s} with positive probability. More precisely, we denote by $\alpha_{i,j}(t, \Theta_{\mathbf{b}}(t))$ the probability with which \mathbf{b}_i makes offer $x_{\mathbf{b}_j}^*$ at time point t given that the buyer's types believed by \mathbf{s} with strictly positive probability are those belonging to $\Theta_{\mathbf{b}}(t)$. Only the probabilities $\alpha_{i,j}(t, \Theta_{\mathbf{b}}(t))$ with $x_{\mathbf{b}_i}^*(t) > x_{\mathbf{b}_j}^*(t)$ and $\mathbf{b}_j \in \Theta_{\mathbf{b}}(t)$ can be non-null. The system of belief is such that, once offer $x_{\mathbf{b}_i}^*(t)$ is observed, all the types \mathbf{b}_j with $x_{\mathbf{b}_j}^*(t) < x_{\mathbf{b}_i}^*(t)$ are removed from $\Theta_{\mathbf{b}}(t)$. Then, the number of possible $\Theta_{\mathbf{b}}(t)$ is linear in the size $\Theta_{\mathbf{b}}(0)$, e.g., if $\Theta_{\mathbf{b}}(0) = \{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$, then the possible $\Theta_{\mathbf{b}}(t)$ are $\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$, $\{\mathbf{b}_2, \mathbf{b}_3\}$, and $\{\mathbf{b}_3\}$. Similarly, the seller's strategy can be represented by probabilities $\beta_j(t, \Theta_{\mathbf{b}}(t))$, i.e., the probability to accept $x_{\mathbf{b}_i}^*(t-1)$ /offer $x_{\mathbf{b}_i}^*(t)$ at t when the buyer's types believed with positive probabilities are $\Theta_{\mathbf{b}}(t)$.

The construction of the game tree is accomplished according to the following rules: 1) no buyer's types makes offer strictly weaker than her optimal offer in

the complete-information game; 2) at time $t > 0$, no agent (buyer and seller) makes offers strictly weaker (w.r.t. her utility function) than the one made by the opponent at the previous time point $t - 1$; 3) at time $t > 0$, no agent (buyer and seller) makes offers that, if accepted at $t + 1$, provide her the same utility she receives by accepting the offer made by the opponent at $t - 1$; 4) no buyer's type makes offers besides $\min\{T_{\mathbf{b}_i}, T_{\mathbf{s}}\}$ and the seller does not make offer besides $\min\{\max\{T_{\mathbf{b}_i}, T_{\mathbf{s}}\}$; 5) at time $t > 0$, an offer x_i is not made if the buyer's type \mathbf{b}_i is out of the game (i.e., $t \geq T_{\mathbf{b}_i}$ or type \mathbf{b}_i has been excluded because the buyer has previously made an offer strictly weaker than the optimal complete-information offer of \mathbf{b}_i).

It can be easily observed that the size of the tree rises exponentially in the length of the deadlines. Differently from what we did for the two-type case, here do not use support-enumeration techniques, but we resort to linear-complementarity mathematical programming. This is because the number of supports rises as 4^n where n is the number of agents' actions, while the space of solutions over which linear complementarity works rises as 2.6^n .

We implemented an *ad hoc* version of the Lemke's algorithm with perturbation as described in [8] to compute a sequential equilibrium. The algorithm is based on pivoting (similarly to the simplex algorithm) where perturbation affects only the choice of the leaving variable. We coded the algorithm in C language by using integer pivoting and the same approach of the revised simplex (to save time during the update of the rows of the tableau). We executed our algorithm with a 2.33 GHz 8 GB RAM UNIX computer. We produced several bargaining instances characterized by the number of buyer's types (from 2 up to 6) and the deadline $T = \min\{\max\{T_{\mathbf{b}_i}, T_{\mathbf{s}}\}$ (from 6 up to 500). Tab. 3 reports the average computational times over 10 different bargaining instances; we denote by '-' the cases whose execution exceeds one hour.

T	number of buyer's types				
	2	3	4	5	6
6	< 0.01 s	0.06 s	0.29 s	3.47 s	929.73 s
8	< 0.01 s	1.32 s	32.94 s	1890.96 s	-
10	< 0.01 s	15.16 s	2734.29 s	-	-
12	< 0.01 s	211.11 s	-	-	-
14	< 0.01 s	3146.20 s	-	-	-
50	0.22 s	-	-	-	-
100	1.55 s	-	-	-	-
500	175.90 s	-	-	-	-

Table 3. Computational times for solving a bargaining game with linear complementarity mathematical programming ($T = \min\{\max\{T_{\mathbf{b}_i}, T_{\mathbf{s}}\}$).

As it can be observed, the computational times are exponential in the bargaining length and have the number of types as basis and only small settings can be solved by using linear-complementarity mathematical programming. Notice that the support-enumeration approach used for the two-types case is much faster than the linear-complementarity approach. This pushes for the development of algorithms for finding approximate solutions.

5 Conclusions and Future Works

The study of strategic bargaining with uncertainty is a challenging game theoretic problem. The literature provides several heuristics-based approaches generally applicable to any uncertain setting, while the optimal approaches work only with very narrow uncertainty settings. In particular, no algorithm works with uncertainty over multiple parameters. In this paper, we focused on one-sided uncertainty. Our main result is the reduction of the bargaining to a finite game. This allows one to resort to well known techniques to solve finite games. We proved that with two types the problem is polynomial (by using support-enumeration techniques), while with more types our algorithm requires exponential time. As a result, only small settings can be solved in exact way. Nevertheless, our reduction allows one to resort to techniques to find approximate equilibria.

In future works, on the one hand, we shall develop algorithms to find an ϵ -approximate equilibrium with a provable bound over ϵ and, on the other hand, we characterize solutions to produce insight over the structure of the problem and design more efficient exact algorithms.

References

1. Chatterjee, K., Samuelson, L.: Bargaining under two-sided incomplete information: The unrestricted offers case. *OPER RES* 36(4), 605–618 (1988)
2. Faratin, P., Sierra, C., Jennings, N.R.: Negotiation decision functions for autonomous agents. *ROBOT AUTON SYST* 24(3-4), 159–182 (1998)
3. Fatima, S.S., Wooldridge, M.J., Jennings, N.R.: On efficient procedures for multi-issue negotiation. In: TADA-AMEC. pp. 71–84. Hakodate, Japan (May 9 2006)
4. Gatti, N., Di Giunta, F., Marino, S.: Alternating-offers bargaining with one-sided uncertain deadlines: an efficient algorithm. *ARTIF INTELL* 172(8-9), 1119–1157 (2008)
5. Gneezy, U., Haruvy, E., Roth, A.E.: Bargaining under a deadline: Evidence from the reverse ultimatum game. *GAME ECON BEHAV* 45, 347–368 (2003)
6. Koller, D., Megiddo, N., von Stengel, B.: Efficient computation of equilibria for extensive two-person games. *GAME ECON BEHAV* 14(2), 220–246 (1996)
7. Kreps, D.R., Wilson, R.: Sequential equilibria. *ECONOMETRICA* 50(4), 863–894 (1982)
8. Miltersen, P.B., Sorensen, T.B.: Computing sequential equilibria for two-player games. In: SODA. pp. 107–116 (2006)
9. Porter, R., Nudelman, E., Shoham, Y.: Simple search methods for finding a Nash equilibrium. In: AAAI. pp. 664–669 (2004)
10. Rubinstein, A.: Perfect equilibrium in a bargaining model. *ECONOMETRICA* 50(1), 97–109 (1982)
11. Rubinstein, A.: A bargaining model with incomplete information about time preferences. *ECONOMETRICA* 53(5), 1151–1172 (1985)
12. Rubinstein, A.: Choice of conjectures in a bargaining game with incomplete information. In: Roth, A.E. (ed.) *Game-Theoretic Models of Bargaining*, pp. 99–114. Cambridge University Press, Cambridge, UK (1985)
13. Sandholm, T.: Agents in electronic commerce: Component technologies for automated negotiation and coalition formation. *AUTON AGENT MULTI-AG* 3(1), 73–96 (2000)

SHORT PRESENTATION

Analysis of Stable Prices in Non-decreasing Sponsored Search Auction

ChenKun Tsung¹, HannJang Ho², and SingLing Lee¹

¹ Dep. of Computer Science and Information Engineering, National Chung Cheng University

² Dep. of Applied Digital Media, WuFeng University
{tck95p, hjho, singling}@cs.ccu.edu.tw

Abstract. Most critical challenge of applying generalized second price (GSP) idea in multi-round sponsored search auction (SSA) is to prevent revenue loss for search engine provider (SEP). In this paper, we propose non-decreasing Sponsored Search Auction (NDSSA) to guarantee SEP's revenue. Each advertiser's bid increment is restricted by minimum increase price (MIP) in NDSSA. The MIP determination strategy influences bid convergence speed and SEP's revenue. Fixed MIP strategy and Additive-Increase/Multiplicative-Decrease (AIMD) principle are applied to determine MIP values, and they are evaluated in this paper. For the convergence speed analysis, fixed MIP strategy converges faster than AIME in most instances. For SEP's revenue, AIMD assists SEP to gain more revenue than fixed MIP strategy by experiments. Simultaneously, SEP's revenue in Vickrey-Clarke-Groves auction (VCG) is the lower bound of that in AIMD.

Key words: Sponsored Search Auction; Generalized Second Price Auction; Minimum Increase Price; Additive-Increase/Multiplicative-Decrease; lower bound

1 Introduction

Recently, search engine provider (SEP) combines advertising and search results on the screen. This kind of advertising application is called as sponsored search auction (SSA).

Many advertisers would like to join SSA due to the pay-per-click design. Only advertiser whose advertisement is clicked by the Internet user is charged. To simulate the click event, the click-through-rate (CTR) is introduced [1]. CTR is a probability that the Internet user clicks on. Thus, the quality of each advertising slot can be estimated by the CTR assumption.

Aggarwal et al. suggest that the CTR should be evaluated according a merchant-specific factor and a position-specific factor [9]. So, the relevance of each advertisement and inputted keywords impacts CTR. For simplification, most related works only consider position-specific factor, such as [1] [2].

Generalized Second price (GSP) [1] is the famous charging function in real world SSA applications. Each clicked advertiser pays equal to the bid value

of next ranked advertiser. Comparing to the idea of paying what he/she bids, advertisers in GSP will save more money.

Bu et al. [7] and Cary et al. [6] study the multi-round SSA, while SEP's revenue may be reduced round by round. When an advertiser is benefited in a worse position, he/she will propose a lower bid in the next round. Therefore, SEP's revenue will be decreased because the revenue comes from the sum of payments.

According to [6] [7], we propose Non-decreasing Sponsored Search Auction (NDSSA) to improve SEP's revenue by allowing bidding on only non-decreasing prices. Thus, advertisers will compete for better slots to improve utilities, that is similar to English auction, and the revenue loss problem is resolved.

However, SEP suffers an extended issue: long-term revenue loss problem. SEP's long-term revenue is the sum of payments after several rounds. Less payment will also improve advertiser's long-term utility. Either the initial bid with extremely low value is proposed or increasing bid values slightly is beneficial for long-term utility of each advertiser. So, SEP's revenue in each round will be raised slowly, and the long-term revenue loss problem is taken place.

All kinds of initial bid values are available in NDSSA. We only focus on solving the second counterattack strategy, increasing bids slowly, by restricting bid increments. The essential bid increment is called as Minimum Increase Price (MIP) in this paper. Each advertiser is allowed to propose only the bid value which is either equal to that in the last round or increased by the MIP value. Thus, advertisers will bid actively due to MIP consideration.

For SEP, the first issue in NDSSA is the convergence speed. After NDSSA begins, each advertiser continuously updates his/her bid value to compete better slot. SEP's revenue is improved during this phase. When no advertiser would like to propose higher bid, NDSSA is converged. Requiring more rounds to reach stable allocation is caused from that bids are increased slowly. So, the convergence speed is an important factor to evaluate the mechanism for SEP.

The second issue is SEP's revenue, and this is most interested by SEP. Since SEP's revenue comes from advertisers' payments, maximizing bid values in each round implies SEP's revenue is improved.

To determine MIP settings, two MIP strategies are proposed: fixed and adaptive MIP strategies. The MIP setting is invariant in each round in fixed MIP strategy. The idea of additive-increase/multiplicative-decrease (AIMD) is applied in adaptive MIP strategy to calculate MIP setting in each round.

Convergence speed and SEP's revenue are discussed in this paper. We proof that fixed MIP strategy converges faster than AIMD in most instances. On the other hand, SEP will obtain more revenue than fixed MIP strategy according to our experiment results.

In the following context, NDSSA is defined in section 2 which includes the mechanism, bidding strategy, and MIP strategies. The convergence speed issue is analyzed in section 3. SEP's revenue comparisons between different MIP strategies are measured by experiments in section 4. The conclusion and future works are shown in section 5.

1.1 Related Work

Most popular payment calculations in SSA are GSP [1] and Vickrey-Clarke-Groves auction (VCG) [2]. Each winner pays bid value ranked in the next slot and the social welfare gap between the winner leaves and joins the auction in GSP and VCG respectively.

Incentive compatibility is the major advantage of VCG. Advertisers are ranked by their advertising valuations, because bidding on other prices is not beneficial for each advertiser. For SEP's revenue and computation cost, VCG is not practical in real word applications [4]. Moreover, VCG is the revenue lower bound of GSP for SEP in some instances [2]. To build a more realistic mechanism, SEP should consider GSP.

Winning the slot to improve the utility is the natural objective of each advertiser. After receiving a satisfied allocation, no advertiser wishes for any deviation, and the auction meets the equilibrium result [5]. Edelman et al. apply the stable idea to define locally envy-free equilibrium [1].

Since the allocation is steady under locally envy-free equilibrium, SEP's revenue is invariant and expectable. Moreover, VCG is the revenue lower bound for SEP when advertisers bid truthfully [2]. Because the steady allocation is the natural target and produces expectable revenue for each advertiser and SEP respectively, winning an envy-free slot is the bidding behavior discussed in NDSSA.

The multi-round assumption is close to the real world instance. Major property in this assumption is that participants will learn from previous result [5]. Cary et al. study the "balance bidding strategy" in the multi-round SSA [7]. Similar to our work, Cary et al. restrict the bid value, but not all instances meet the steady allocation. The outcome stability is important for SEP due to the revenue expectation, so the stability is considered in NDSSA.

Restricting minimum bid prices has the same effect with MIP. Even-Dar et al. modified Tâtonnement process to compute the minimum bid value [8]. When applying the idea of Even-Dar et al., SEP will gains more revenue than VCG. If the auction efficiency is guaranteed, the mechanism is more useful for SEP.

2 NDSSA

SEP must solve the revenue loss problem when applying GSP in multi-round SSA. Consider the advertiser occupied 1^{st} slot, for example. If he/she is benefited in the 2^{nd} slot, he/she will propose a lower bid price for moving to 2^{nd} slot in the next round. The advertiser is benefited by payment decrease, but SEP revenue is reduced simultaneously.

2.1 Auction Mechanism

An NDSSA instance includes an SEP, and $k + 1$ advertisers that compete for k advertising slots. Suppose that each advertiser is interested in the same keyword and has the ability to update his/her bid value in each round. Payments are

calculated by GSP, i.e. $p_i^x = b_{i+1}^x$, where p_i^x is ad_i 's payment in x^{th} round and b_{i+1}^x is ad_{i+1} 's bid in x^{th} round.

Each advertiser ad_i has two parameters: the valuation and the initial bid rate (IBR) IBR_i . The valuation v_i is the worth per each click, and IBR indicates the ratio of valuation to the initial bid value. So the initial bid value is $v_i \times IBR_i$.

In x^{th} round, each advertiser is allowed to propose two kinds of bids b_i^x .

1. same bid value, i.e. $b_i^x = b_i^{x-1}$, or
2. higher value, i.e. $b_i^x \geq b_i^{x-1} + MIP^{x-1} + \epsilon, \forall \epsilon \geq 0$.

where all advertisers obey the same MIP value in each round.

Each slot sl_j has a click probability, called click-through rate (CTR), to simulate the slot importance. Without Lost of Generality, the better slot has higher CTR value. Therefore, ad_i allocated in sl_j pays $b_y^x \times CTR_j$ expectedly, where ad_y is the winner of sl_{j+1} .

2.2 Bidding Strategy

Consider ad_i occupies sl_j , the utility in x^{th} round is denoted by $u_i^x(j) = CTR_j \times (v_i - p_i^x)$. We only consider *Rational Bidding* advertiser in this paper. This implies no advertiser will bid higher than his/her valuation, i.e. $b_i^x \leq v_i$.

According to the concept of locally envy-free equilibrium, b_i^x will be increased only when sl_{j-1} is more beneficial than sl_j , i.e. $u_i^x(j-1) > u_i^x(j)$. Thus, ad_i will bid $\min\{(b_y^x + 1), (b_i^x + MIP)\}$, where ad_y is ranked in sl_{j-1} .

2.3 MIP Strategies

The MIP value of fixed MIP strategy determines SEP's revenue. For higher settings, higher bid increment will limit the final bid value. Advertisers can not bid close to their valuations, so SEP's revenue in higher MIP setting may be less than in lower MIP setting. Consider the advertiser with valuation 50, bid value 40, and MIP 11, for example. The advertiser must propose \$51 at least. According to rational bidding, SEP will lose \$10 at most.

AIMD is used to probing unknown bandwidth in a TCP connection [3]. We apply the adjustability of AIMD to determine the MIP setting in each round. No bid update indicates the congestion in TCP, so the MIP value is set to one half. Otherwise, MIP is increased by one continuously. When each advertiser keeps the same bid under $MIP = 1$, advertisers have no idea to increase bids, and NDSSA converges.

To maximizing SEP's revenue, AIMD requires more rounds than fixed MIP strategy to check that NDSSA converges or not. SEP has the trade-off between the convergence speed and the revenue for determining MIP strategies.

3 Convergence Speed Analysis

3.1 Fixed MIP Strategy

SEP requires determining the MIP value before NDSSA begins. The MIP setting is invariant throughout the auction. In the worst case, Theorem 1 shows the number of rounds required to converge by fixed MIP strategy.

Theorem 1. *Consider ad_{mab} has maximum available bid amount over all advertisers in an NDSSA instance, where MIP^0 is initial MIP setting and $mab = \arg \max_{v_i \neq 1} v_i(1 - IBR_i)$. In the worst case, the number of rounds r^F required to meet the stable allocation by fixed MIP strategy is as follows.*

$$r^F = \lceil \frac{v_{mab}(1 - IBR_{mab})}{MIP^0} \rceil$$

Proof. This proof is divided into two portions. We first deal with why the advertiser ad_{mab} dominates the convergence speed and then calculate the number of convergence rounds.

The advertiser ad_{mab} , where $mab = \arg \max_{v_i \neq 1} v_i(1 - IBR_i)$, represents that he/she has most available prices to bid. In other words, ad_{mab} still can increase his/her bid value while others meet their valuations. The advertiser with highest valuation is excluded, because he/she will win the 1st slot when bidding over 2nd ranked advertiser rather than his/her valuation. Therefore, ad_{mab} , except for ad_1 , dominates the convergence bottleneck in fixed MIP strategy.

We have the maximum available bid increment $v_{mab}(1 - IBR_{mab})$, and the increment divided by the MIP setting is the number of convergence rounds required in the worst case.

3.2 AIMD

The convergence speed of AIMD is analyzed by two portions. The first part is the first decrease of MIP value, and the second portion is the remainder rounds. They are shown in Lemma 1 and 2.

Lemma 1. *In the worst case of the NDSSA with AIMD, MIP^{h+1} will be decreased, where $h = \sqrt[2]{(MIP^0)^2 + 2v_{mab}} - MIP^0$.*

Proof. Suppose the MIP value is decreased at $(h + 1)^{th}$ round. All bids in h^{th} and $(h + 1)^{th}$ rounds are the same, i.e. $b_i^h = b_i^{h+1}$. In Fig. 1, the bid value is $b_{mab}^h + MIP^h$ where $MIP^h = MIP^0 + h$, and the pink area indicates the sum of bid increments in the auction, that is $h \times (MIP^0 + (MIP^0 + h))/2$.

If ad_{mab} still increases his/her bid in $(h + 1)^{th}$ round, he/she will overbid, i.e. $b_{mab}^h + MIP^h \geq v_{mab}$. The first round of MIP decrease h can be derived.

$$\begin{aligned} b_{mab}^h + MIP^h &\geq v_{mab} \\ h \times (MIP^0 + (MIP^0 + h)) &\geq 2v_{mab} \\ h^2 + 2MIP^0h - 2v_{mab} &\geq 0 \end{aligned}$$

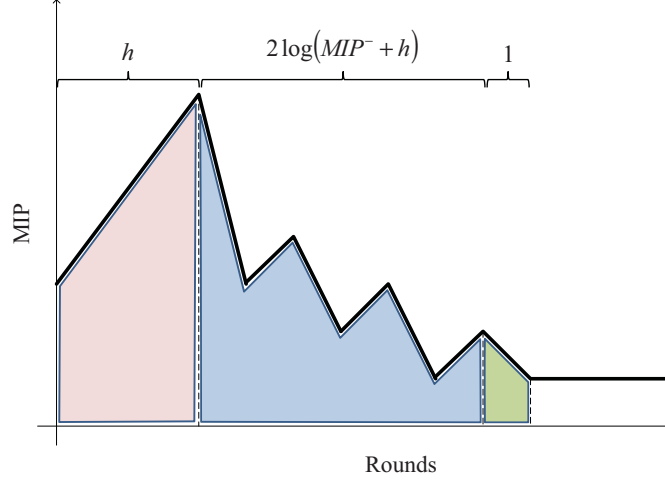


Fig. 1. the MIP modification history of NDSSA in the worst case

$$\begin{aligned}
 h &\geq \frac{-2MIP^0 \pm \sqrt{4(MIP^0)^2 + 8v_{mab}}}{2} \\
 &= \sqrt[2]{(MIP^0)^2 + 2v_{mab}} - MIP^0
 \end{aligned}$$

According to Lemma 1, we derive that higher initial MIP settings result in faster convergence. Then, we focus on the remainder rounds in the worst case.

Lemma 2. *After first MIP decrease, NDSSA with AIMD requires $2\lceil \log(MIP^0 + h) \rceil$ rounds to converge, where $h = \sqrt[2]{(MIP^0)^2 + 2v_{mab}} - MIP^0$ and $mab = \arg \max_{v_i \neq 1} v_i(1 - IBR_i)$ in the worst case.*

Proof. As shown in Fig. 1, the available bid increment is at most $MIP^0 + h$ in $(h + 1)^{th}$ round. If the assumption is false, $MIP^0 + h + 1$ for example, ad_{mab} is able to update his/her bid in $(h + 1)^{th}$ round due to $MIP^{h+1} = (MIP^0 + h) + 1$.

The remainder rounds in the worst case is composed of decrease-increase pairs. Consider the idea in $(h + 2)^{th}$ round. We have $MIP^{h+2} = MIP^{h+1}/2 = (MIP^0 + h + 1)/2$ and the available bid increment is $MIP^0 + h$. So ad_{mab} increases his/her bid by $(MIP^0 + h + 1)/2$. The remainder bid increment is $(MIP^0 + h) - (MIP^0 + h + 1)/2 = (MIP^0 + h - 1)/2$ in the first round of decrease-increase pair, and the MIP value should be increased by one. Because the MIP value is higher than the available bid increment, e.g. $((MIP^0 + h + 1)/2) + 1 \geq (MIP^0 + h - 1)/2$, no bid update will be taken place in the second round of decrease-increase pair.

The NDSSA with AIMD will decrease the MIP setting to one half, and each decrease contains a pair of rounds. When $MIP = 1$, each advertiser will no longer update his/her bid value. Therefore, the remainder rounds is $2\lceil \log(MIP^h) \rceil = 2\lceil \log(MIP^0 + h) \rceil$.

Combining Lemma 1 and 2, the total number of convergence rounds required by the NDSSA with AIMD is shown in Theorem 2.

Theorem 2. *NDSSA with AIMD will converge at most $r^A = h + 2\lceil \log(MIP^0 + h) \rceil$ rounds, where $h = \sqrt[2]{(MIP^0)^2 + 2v_{mab}} - MIP^0$ and $mab = \arg \max_{i \neq 1} v_i(1 - IBR_i)$.*

The upper bound of convergence rounds for fixed MIP strategy and AIMD are shown in Theorem 1 and 2 respectively. Now, we are analyzing the condition that fixed MIP strategy converges faster than AIMD.

Theorem 3. *When the MIP value is decreased after $(r^F - 2 \log v_{mab})$ rounds in AIMD, fixed MIP strategy converges faster than AIMD, where $r^F = \lceil v_{mab}(1 - IBR_{mab})/MIP^0 \rceil$, in the worst case.*

Proof. Combining Theorem 1 and 2, assume that fixed MIP strategy converges faster than AIMD, i.e. $r^F \leq r^A$. The objective is to proof $h \geq r^F - 2 \log v_{mab}$.

$$\begin{aligned}
r^F &\leq r^A \\
&= h + 2 \log(MIP^0 + h) \\
&\leq h + 2 \log\left(\frac{v_{mab}}{2}\right) + 2 \\
&= h + 2 \log v_{mab} \\
\Rightarrow h &\geq r^F - 2 \log v_{mab}
\end{aligned}$$

4 Simulation

The distributions of valuation, CTR, and IBR, are shown in table 1, 2, and 3 respectively. The gaps of valuation and CTR settings include uniform, linear, exponential increasing, exponential decreasing, and random. The IBR settings are not required to restrict as a decreasing order, so the previous advertiser may have smaller IBR value than the next one. Since the maximum valuation is 50, initial MIP values are evaluated from 1 to 50 for each instance. So, we have 6250 measurements.

Following experiments are evaluated in this paper: (1) robustness, comparing which mechanism produces more SEP's revenue in more instances, (2) overall SEP's revenue comparison, evaluating SEP's revenue for all mechanisms under stable allocations, (3) SEP's average revenue, analyzing SEP's average revenue after converging, and (4) SEP's long-term revenue comparison, discussing the SEP's total revenue during a specific round.

4.1 Robustness

For two mechanisms x and y , we say that x is more robust than y if the number of instances with more SEP's revenue in x is more than that in y .

Table 1. valuation setting

case #	ad_1	ad_2	ad_3	ad_4	ad_5
1	50	40	30	20	10
2	50	34	22	14	10
3	50	46	38	26	10
4	50	45	38	26	10
5	44.46	41.00	40.68	28.80	26.96

Table 2. CTR setting

case #	sl_1	sl_2	sl_3	sl_4
1	0.8	0.6	0.4	0.2
2	0.8	0.376	0.164	0.053
3	0.8	0.747	0.641	0.429
4	0.8	0.76	0.4	0.38
5	0.8	0.598	0.475	0.39

The pairwise comparisons of fixed MIP strategy, AIMD, and VCG are evaluated in this experiment, and results are shown in Figure 2, 3, and 4. The labels $x > y$, $x = y$, and $x < y$ in each figure denote the number of instances that SEP’s revenue of mechanism x is more than, same to, and less than mechanism y respectively.

In Figure 2, AIMD is more robust than fixed MIP strategy. After initial MIP 46, SEP gains more revenue in AIMD than in fixed MIP strategy in all instances. Recall our claim in suction 2.3: higher MIP values will increase the gap between stable bid value and the valuation for fixed MIP strategy. The conjecture is confirmed in this measurement.

The robustness comparison between AIMD and VCG is shown in Figure 3. AIMD is also more robust than VCG. SEP’s revenue in VCG is the lower bound of that in GSP just in some instances [1]. The revenue lower bound of GSP is extended to all instances in NDSSA according to our simulation.

Figure 4 draws the comparison between fixed MIP strategy and VCG. As the initial MIP increases, fixed MIP strategy performs worse and worse. The disadvantage of fixed MIP strategy, less SEP’s revenue in higher MIP settings, is explored clearly in this simulation. Under lower MIP settings, SEP is benefited, and fixed MIP strategy is more robust than VCG in average.

4.2 Overall SEP’s Revenue Comparison

Fig 5 shows the overall comparison about SEP’s revenue between AIMD, fixed MIP strategy, and VCG. Each square includes three comparison results: “>”, “=”, and “<”, and they stand for how many instances that the left mechanism is better than, equal to, and worse than above one respectively. The VCG-ALL comparison, for example, represents that VCG produces more SEP’s revenue than all mechanisms in 892 instances, identical to 98 instances, and less than in 11510 instances.

Table 3. IBR setting

case #	ad_1	ad_2	ad_3	ad_4	ad_5
1	0.9	0.7	0.5	0.3	0.1
2	0.1	0.3	0.5	0.7	0.9
3	0.9	0.3	0.7	0.1	0.5
4	0.5	0.1	0.7	0.3	0.9
5	0.87	0.36	0.57	0.10	0.92

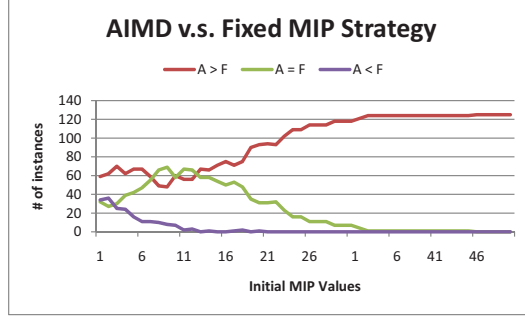


Fig. 2. SEP’s revenue comparison between AIMD and fixed MIP strategy in the stable allocation.

AIMD produces more revenue in 11134 instances (89.07%) approximately, and only 192 instances (1.54%) are worse than other mechanisms. The second one is the NDSSA with fixed MIP strategy, and last one is VCG. Comparing to VCG, AIMD improves 990 instances (15.84%) of fixed MIP strategy. Therefore, SEP’s revenue of the stable allocation is maximized by adopting AIMD in most instances.

4.3 SEP’s Average Revenue

In this simulation, we deal with the impact of initial MIP values on SEP’s revenue in the stable allocation. Figure 6 shows the result. The x-axis indicates initial MIP values, and the y-axis is SEP’s revenue averaged by all instances under the same MIP value.

Since the payment of VCG is calculated according to valuations, SEP’s revenue is identical to various initial MIP values. In average, AIMD and fixed MIP strategy perform better than VCG.

AIMD receives a more stable result than fixed MIP strategy, and the impact of initial MIP values is slight. Since AIMD adjust MIP values in each round, bid values are optimized in average.

SEP’s revenue of fixed MIP strategy decreases slightly in lower initial MIP settings. After initial MIP is 18 approximately, SEP’s revenue drops dramatically until MIP 46. Increasing initial MIP value implies that the gap between the valuation and the stable bid value becomes larger. Therefore, SEP’s revenue is

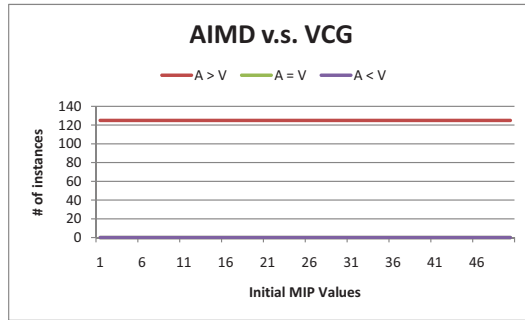


Fig. 3. SEP’s revenue comparison between AIMD and VCG in the stable allocation.

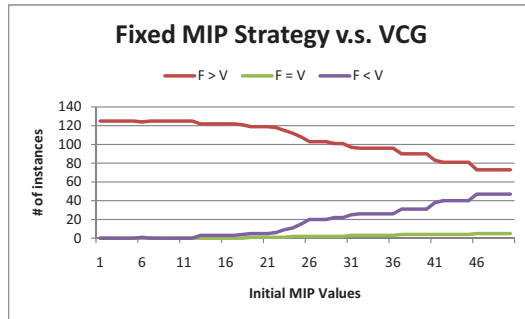


Fig. 4. SEP’s revenue comparison between fixed MIP strategy and VCG in the stable allocation.

decreased. Lower initial MIP values are better for SEP’s revenue in the stable allocation.

4.4 SEP’s Long-term Revenue Comparison

Given a maximum round, the long-term revenue of SEP is the sum of revenue in each round. Only AIMD and fixed MIP strategy is compared in this simulation, and the result is shown in Figure 7.

Given different maximum rounds, the variance degree of SEP’s long-term revenue is not different too much for both AIMD and fixed MIP strategy. Similar to Figure 6, initial MIP values almost do not vary SEP’s total revenue in AIMD. SEP gains less revenue when initial MIP value increase in fixed MIP strategy. In Figure 7(a), SEP gains more revenue in fixed MIP strategy than AIMD in a few instances. As the number of maximum rounds increases, AIMD performs better in all initial MIP values. AIMD is more appropriate than fixed MIP strategy for the long term revenue for SEP.

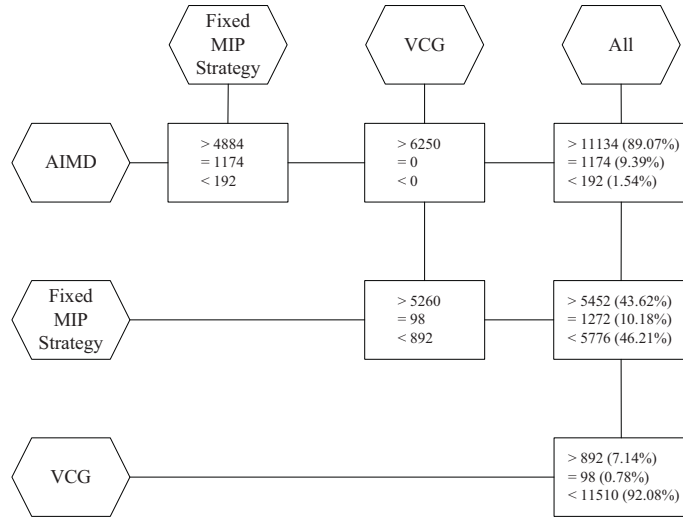


Fig. 5. Overall SEP's revenue comparison between all mechanisms.

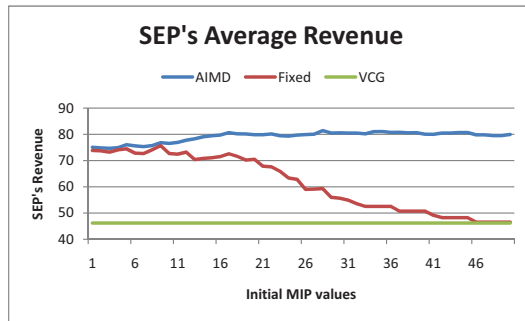
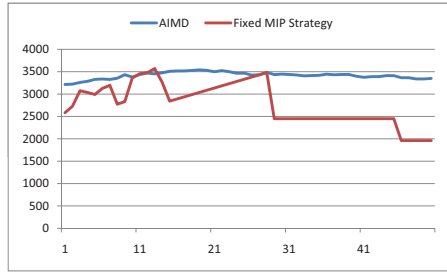


Fig. 6. SEP's average revenue comparison under different initial MIP values.

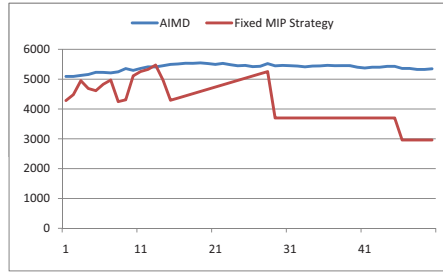
5 Conclusion

When applying GSP to a multi-round SSA, SEP suffers the revenue loss problem. We propose Non-decreasing Sponsored Search Auction (NDSSA) to solve this problem while each advertiser is allowed to propose only non-decreasing bids in next round. Minimum Increase Price (MIP) is used in NDSSA to control the bid value for improving long term revenue.

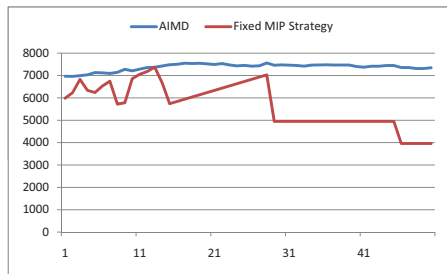
Fixed MIP strategy and AIMD are applied to compute MIP values. For theoretical convergence speed analysis, fixed MIP strategy converges faster than AIMD in most instances. For SEP's revenue comparison of our simulations, AIMD not only produces better but is more robust than fixed MIP strategy. Thus, fixed MIP strategy is outstanding in short-term plan, and AIMD is for long-term consideration.



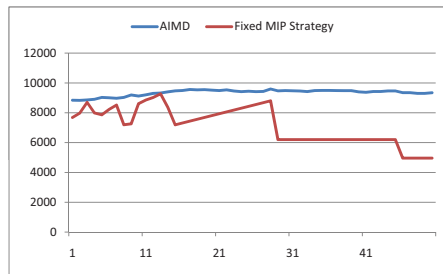
(a) After 50 rounds.



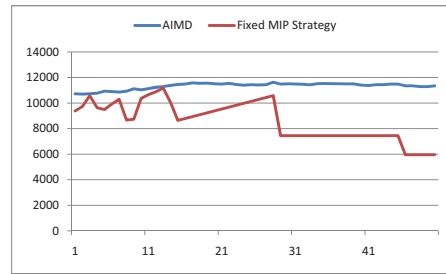
(b) After 75 rounds.



(c) After 100 rounds.



(d) After 125 rounds.



(e) After 150 rounds.

Fig. 7. SEP's total revenue comparison for given different rounds.

SEP's revenue is improved in NDSSA in this paper. However, SEP has no idea to capture advertiser's satisfaction. If the expected objectives, the utility for example, are not achieved, advertisers may leave the auction. SEP's revenue will also be decreased potentially. Therefore, measuring satisfactions for any participant will be studied in the future.

Acknowledgement

This work was supported in part by Taiwan NSC under grant no. NSC 98-2221-E-274-006 and NSC 99-2221-E-274-007. The author like to thank reviewers for their insightful comments which helped to significantly improve the paper.

References

1. B. Edelman, M. Ostrovsky, and M. Schwarz, "Internet Advertising and the Generalized Second Price Auction: Selling Billions of Dollars Worth of Keywords," *American Economic Review*, Vol. 97(1), pp. 242-259, March 2007.
2. H. R. Varian, "Position Auction," *International Journal of Industrial Organization*, Vol. 25(6), pp. 1163-1178, December 2007.
3. D-M Chiu and R. Jain, *Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks*, *Computer Networks and ISDN Systems*, vol. 17, no. 1, 1989, pp. 1-14.
4. M. H. Rothkopf, *Thirteen Reasons Why the Vickrey-Clarke-Groves Process Is Not Practical*, *Operations Research*, Vol. 55, no. 2, 2007, pp. 191-197.
5. N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*, Cambridge University Press, 2007.
6. T. M. Bu, X. Deng, and Q. Qi, *Forward looking Nash equilibrium for keyword auction*, *Information Processing Letters*, Vol. 105, pp. 41-46, 2008.
7. M. Cary, A. Das, B. Edelman, I. Giotis, K. Heimerl, A. R. Karlin, C. Mathieu, and M. Schwarz, *Greedy Bidding Strategies for Keyword Auctions*, *Proceedings of the 8th ACM Conference on Electronic Commerce (EC-07)*, pp. 262-271, 2007.
8. E. Even-Dar, J. Feldman, Y. Mansour, and S. Muthukrishnan, *Position Auctions with Bidder-Specific Minimum Prices*, *Proceedings of the 4th International Workshop on Internet and Network Economics (ED08)*, pp. 577-584, 2008.
9. G. Aggarwal, A. Goel, and R. Motwani *Truthful Auctions for Pricing Search Keywords*, *Proceedings of the 7th ACM Conference on Electronic Commerce (EC-06)*, pp. 1-7, 2006.